Introduction

Assumptions of this Training Manual

Since the introduction of the microcomputer into homes and schools in the early 1980's teachers have faced the challenge of integrating new technology into the classroom. For some this meant seemingly endless workshops and inservice days. For others it meant dodging the issue and avoiding one more electronic gadget. For some there was the hope that it was a fad that would go away.

Meanwhile parents and school boards became convinced that they had to "do something" about their schools. For many that meant spending money on computers as the answer to education's ills. As a result computers, several millions of them as of January 1989, have been placed in classrooms and libraries. Teachers have been asked to figure out how they should be used.

This manual is based on the assumption that the most important use of computers in schools is the use students make of them, rather than the use teachers make of them. We assume that it is the teacher's responsibility to point the way. It is the student's responsibility to walk the path and assemble the knowledge for themselves.

HyperCard in Education

Module 1 Participant's Guide

Joseph F. Hofmeister • Joyce B. Rudowski
The HyperSchool
Cincinnati Country Day School
6905 Given Road, Cincinnati, Ohio 45243
513-561-7298 AppleLink K0197

Module 1

Introducing HyperCard

Assumptions of the Course
Evaluate Mac Literacy

Some of you may be uncomfortable with the Mac at the start of this session. Everyone should have at least worked on a Mac prior to this training session's start but a given is that there are some people who have very little experience. Perhaps you have created a document on MacWrite, for example, and have printed it out, but their understanding of the desktop concept, how things are saved, what different icons mean, how you would copy a program or stack from one disk to another is fuzzy at best. **(fig. 1)**

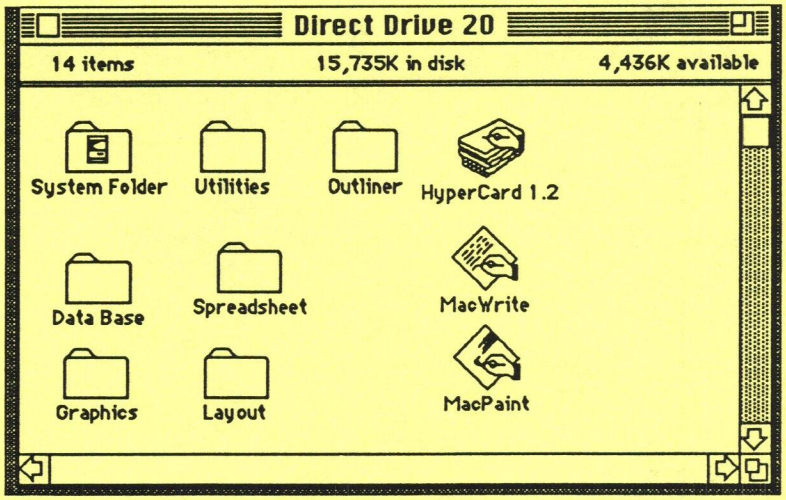


fig. 1 The Desktop is the first introduction to the Macintosh. The desktop contains icons of the applications and files available on the disk. Notice that some of the icons look like small file folders. They are folders which can hold other folders, applications and files.

Desktop	Finder	Menus	Icons	Select
The first screen of information on the Mac, the directory of icons.	The system software which organizes files and programs.	The pull down arrangement of choices on the Mac.	The pictures which represent programs and files on the Mac.	Clicking the mouse cursor while pointing to an icon or words.

Introducing the HyperCard Environment Begin with the HOME Card

The way that we handle this problem is not to go back to square one and turn module one into a Mac training session, but rather to leapfrog over it by identifying the HyperCard environment as THE environment. **(fig. 2)**

Notice the vocabulary words at the bottom of each page. They are intended to help bridge the gap of what you know about the Macintosh and the vocabulary of HyperCard.

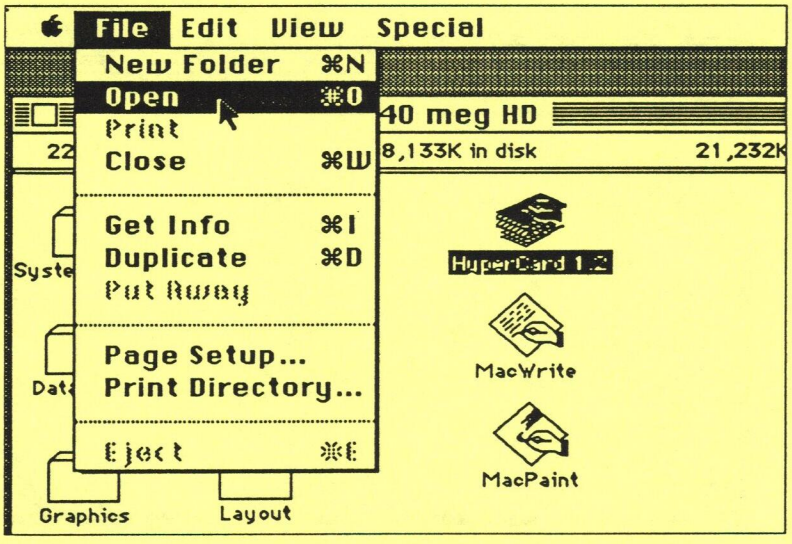


fig. 2 The Home card is the place to begin and to return often in HyperCard. The icons on the Home card are buttons to other cards and stacks.

Stack	Home	Card	Tools
A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.	Home is a stack designated as the starting point for HyperCard.	A card is a screenful of information, one page of a stack.	Tools provide the power to build cards and stacks.

HyperCard as THE Environment

Since HyperCard is a subset of the Mac environment, we train to it and let the Mac literacy occur as a byproduct - kind of like taking French language students and putting them in a situation where they had to be so concerned about what they had to get done that they forget that they have to do it in French and just get on with it. Begin to speak in the language of HyperCard and you will soon accept it as your own. **(fig. 3)**

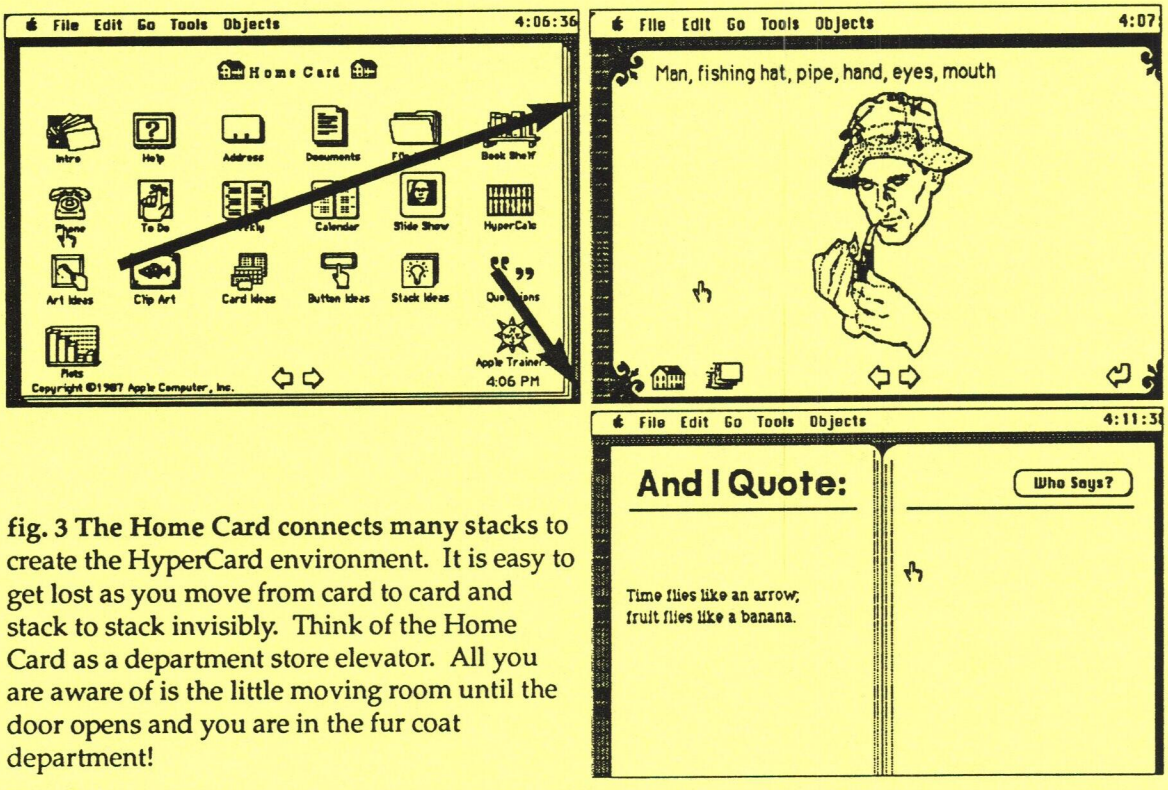


fig. 3 The Home Card connects many stacks to create the HyperCard environment. It is easy to get lost as you move from card to card and stack to stack invisibly. Think of the Home Card as a department store elevator. All you are aware of is the little moving room until the door opens and you are in the fur coat department!

Menus	Icons	Select	Stack
The pull down arrangement of choices on the Mac.	The pictures which represent programs and files on the Mac.	Clicking the mouse cursor while pointing to an icon or words.	A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.

Discuss the Buttons and Cards of Home

The HyperCard Home stack is the first thing that is shown to the participants in the class. This is the Home stack as it comes from the box, the original Home with all the icons on it.

This picture may seem overwhelming at first. Ironically the original Home is a very easy place to get lost.

It's easy to get lost!

Home Card	Home Stack	Tools	Objects	Button
Home is a card designated as the starting point for HyperCard.	The Home Stack is the group of cards that operate HyperCard.	Tools provide the power to build cards and stacks.	The 5 objects of HyperCard are the basis of stack creation.	One of the 5 objects, buttons can be programmed at will.

Explain the Home Card as an Elevator

Many, many teachers get this far and are lost forever because they click on things, disappear into the electronic mists and never return. It can be helpful to describe the Home Card as a department store elevator. **(fig. 4)**

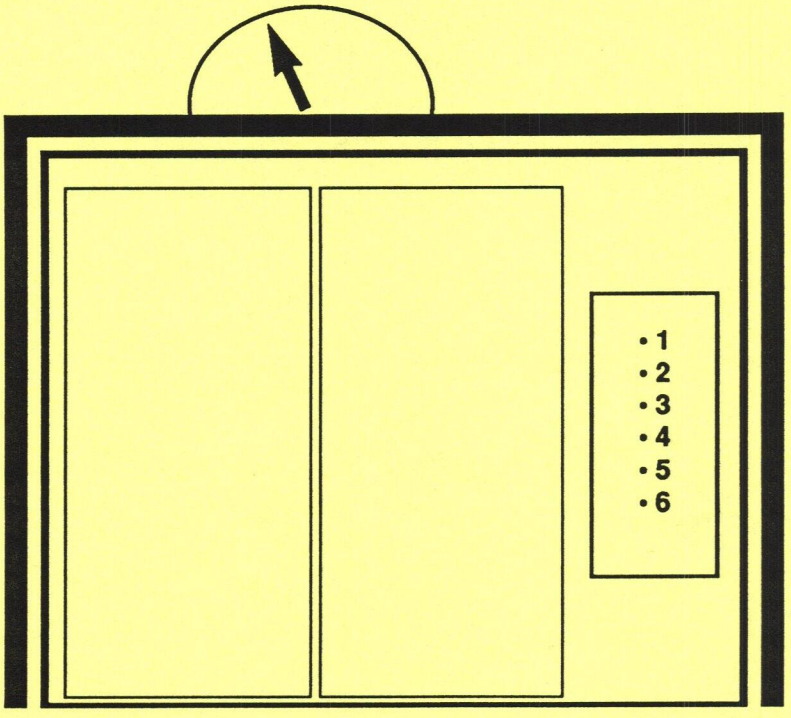


fig. 4 The Home Stack is the elevator which transports us to the exciting and varied stacks connected to the Home Card by buttons. Each stack offers a variety of information as well as connections to yet other stacks. One can always return to the Home Card and continue to other stacks.

Menus	Icons	Stack	Home
The pull down arrangement of choices on the Mac.	The pictures which represent programs and files on the Mac.	A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.	Home is a card designated as the starting point for HyperCard.

The Home Card as Elevator

When we step into an elevator all we see can is the small room we are standing in. However, by pushing the buttons and waiting for the door to open we are transported to the lingerie department, or men's wear or the toy shop! We can stay in that department an indefinite amount of time. There is lots of activity, sound and motion on each floor, just as there is lots to see and do in each stack and on each card. When the shopping is completed we can step back into the elevator (return to the Home Card) and push another button.

It is possible to leave the department by another route, say by walking the stairs to another floor or leaving the building altogether. In any case the elevator is always available for a safe return to familiar territory or ready to take us on some new adventure.

Card	Tools	Objects	Button	Non Linear
A card is a screenful of information, one page of a stack.	Tools provide the power to build cards and stacks.	The 5 objects of HyperCard are the basis of stack creation.	One of the 5 objects, buttons can be programmed at will.	A style of thought which creates irregular paths to learning.

You May Feel Confused

Don't feel hopeless if you are confused at first about the Home Card. The Home Card is, perhaps, the best example of non-linear organization of non-parallel applications. Bill Atkinson, the author of *HyperCard*, has included nearly everything important that anyone could have thought of, all jammed on one card - a powerful, mind-expanding place to be.

Icons	Stack	Home	Card
The pictures which represent programs and files on the Mac.	A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.	Home is a card designated as the starting point for HyperCard.	A card is a screenful of information, one page of a stack.

HyperCard makes Connections
The Heart of the Matter

The Home card is the common denominator, the place where we begin when we first open the boxed product. Looking at the Home Card is an excellent place to get a first glimpse of what HyperCard really is - a way of making connections. **(fig. 5)**

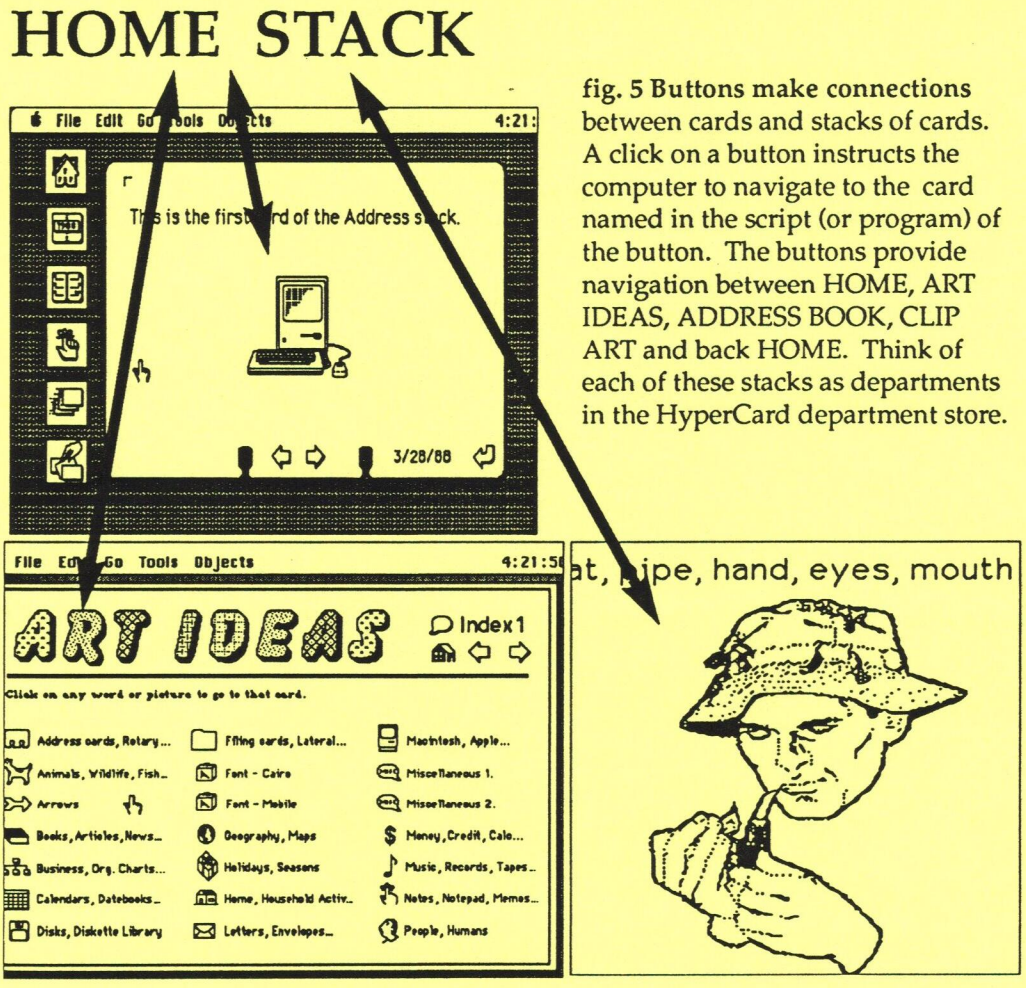


fig. 5 Buttons make connections between cards and stacks of cards. A click on a button instructs the computer to navigate to the card named in the script (or program) of the button. The buttons provide navigation between HOME, ART IDEAS, ADDRESS BOOK, CLIP ART and back HOME. Think of each of these stacks as departments in the HyperCard department store.

Tools	Objects	Button	Browse	Non Linear
Tools provide the power to build cards and stacks.	The 5 objects of HyperCard are the basis of stack creation.	One of the 5 objects, buttons can be programmed at will.	To browse in a stack means to look at it casually, like a magazine.	A style of thought which creates irregular paths to learning.

Navigate to and from the Home Card
Don't try to Explain Everything
Stress the Connections made

We will begin on the Home Card and navigate to several different stacks. Follow the leader and mimic the movement from one stack to another. Think about the idea of "making connections" between stes of information. Don't worry about understanding everything. It will be a great success if the notion that cards can be connected is clear after this exercise.

Stack	Home	Card	Tools
A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.	Home is a stack designated as the starting point for HyperCard.	A card is a screenful of information, one page of a stack.	Tools include graphics and text tools to manage both media.

Continue to Navigate among Cards and Stacks

Spend time navigating through several cards on each stack visited. It should become clear that you are going from card to card and that you are now reconnecting to your Home Card. The ability to navigate through HyperCard is critical to a comfortable use of stacks. It is not possible to spend too much time practicing navigation. At the same time you will be learning the vocabulary of HyperCard by identifying cards, buttons, text fields, menus and the cursor.

Tools	Objects	Buttons	Browse	User Levels
Tools provide the power to build cards and stacks.	The 5 objects of HyperCard are the basis of stack creation.	One of the 5 objects, buttons can be programmed at will.	To browse in a stack means to look at it casually, like a magazine.	Five levels in HyperCard afford more tools as the level increases.

Introduce the 5 User Levels
A Brief Explanation

Navigate your way back to the Home card and move to the last card on the stack. **(fig. 6)** This card lists the different levels that a person can work in while using this program. The difference between the browsing level and the scripting level is the increased power of the scripting level. The underlying message is: you don't have to have a degree in computer science to run this program; anyone can get in, at least on the browsing level. During the training sessions you will be working at the scripting level which affords maximum power.

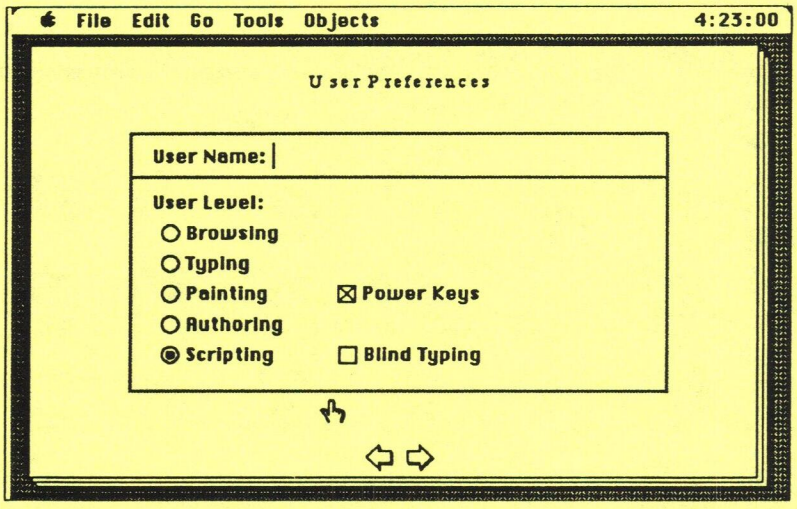


fig. 6 User Levels may be changed on the last card of the Home Stack. Each level adds functions to HyperCard.

Cursor	Home	Card	Icons	Tool
The pointer of the mouse changes shape depending on the tool used.	HyperCard uses Home as the center of navigation.	A card is a screenful of information, one page of a stack.	Pictures or icons can be visual clues to the action of a button.	Tools include graphics and text tools to manage both media.

The Idea of Navigating to Emphasize the Connections of HyperCard

Begin again at the Home card and think about the concept of a stack as a pile of 3x5 cards. You get around in a stack by navigating from one card to another. Look at the choices on the **Go** menu. Each of the choices is used to navigate from the Home card. **(fig. 7)**

Starting with the menu for navigation gives you something to hold on to if you can't find buttons. Furthermore, simple navigational buttons are more keenly appreciated when you have been limited to menu pull downs to get around in a stack.

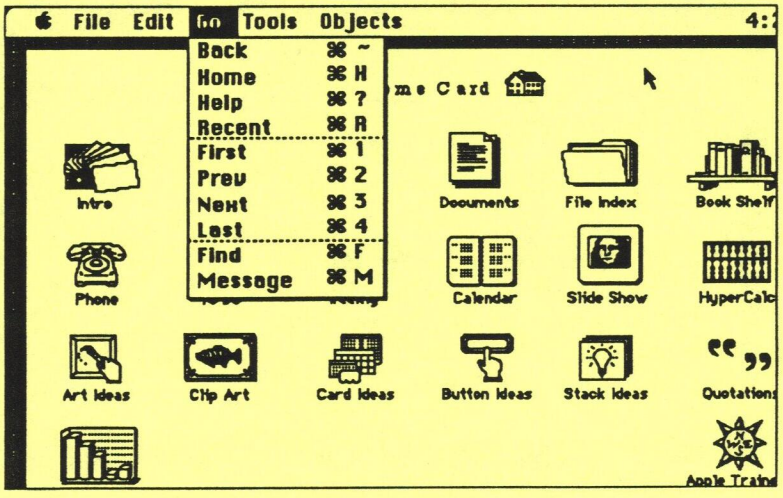


fig. 7 The Go menu provides immediate navigation. When buttons are not apparent, the Go menu is a means to move between cards and stacks. The first four choices allow easy movement between stacks, the second four choices provide easy movement between cards.

Objects	Button	Browse	User Levels	Navigate
The 5 objects of HyperCard are the basis of stack creation.	One of the 5 objects, buttons can be programmed at will.	To browse in a stack means to look at it casually, like a magazine.	Five levels in HyperCard afford more tools as the level increases.	Navigation is the movement around stacks by buttons or menu.

Explain the Process for a 'Follow-Me' Event Step-by-Step Stack Building

A "Follow-Me" exercise in this training manual refers to an exercise where a leader performs the task on a big screen display or large monitor and you follows along, step by step, at your own computer. This technique will be used often as you increase your familiarity with HyperCard.

Stack	Home	Card	Tools
A stack is a file created by HyperCard. The stack icon looks like a stack of cards. Stacks are created, named and automatically saved on a disk.	Home is a stack designated as the starting point for HyperCard.	A card is a screenful of information, one page of a stack.	Tools include graphics and text tools to manage both media.

Begin Together at the Home Card

In this Follow-Me, pull down the **File** menu, go to **New Stack** and stop at the dialog box that comes up. **(fig. 8a-d)** Take your time finding New Stack, which should be darkened to indicate that it is the selected option.

Now you will name the stack by typing in the name. Everyone will call this the First Stack.

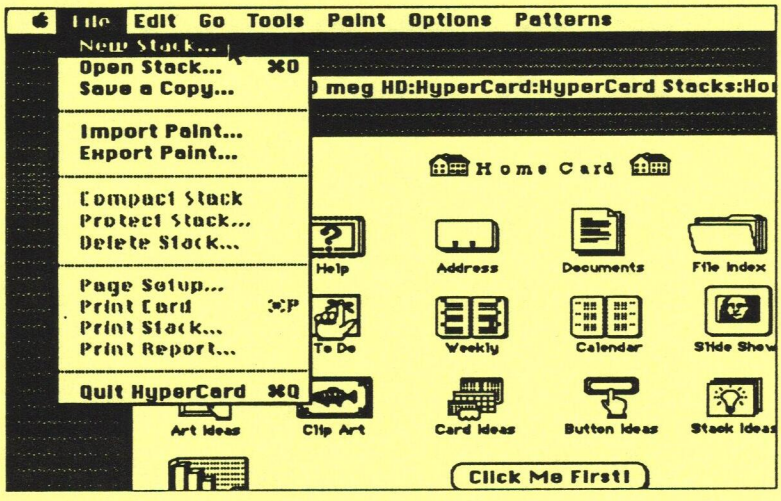


fig. 8a Follow me: New Stack
Drag the mouse on the **File menu** to choose **New Stack.** Let up the mouse and a dialog box will appear.

Objects	Button	Browse	User Level	Navigate
The 5 objects are the stack, background, card, button and field.	One of the 5 objects, buttons can be programmed at will.	To browse in a stack means to look at it casually, like a magazine.	Five levels in HyperCard afford more tools as the level increases.	Navigation is the movement around stacks by buttons or menu.

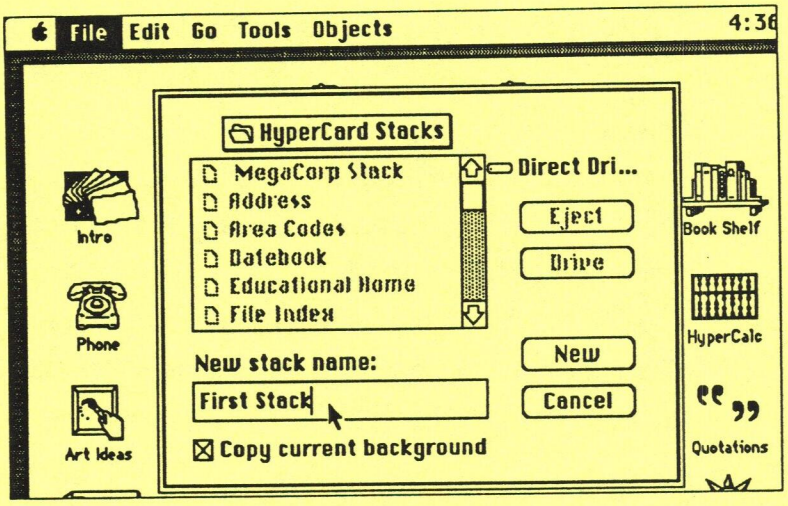


fig.8b Name the stack: First Stack. Type the name into box marked "New stack name".

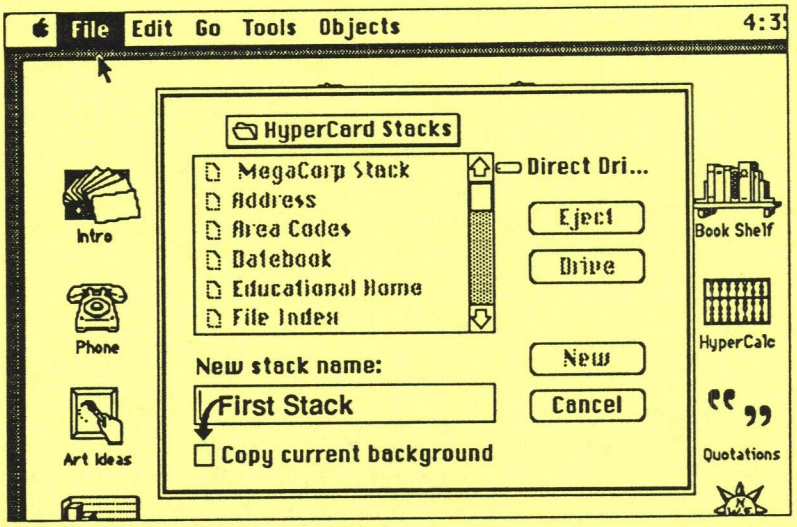


fig. 8c Click to <u>unselect</u> the current background. HyperCard copies the appearance of the current stack unless you clear the X out of the "Copy Current Background" box.

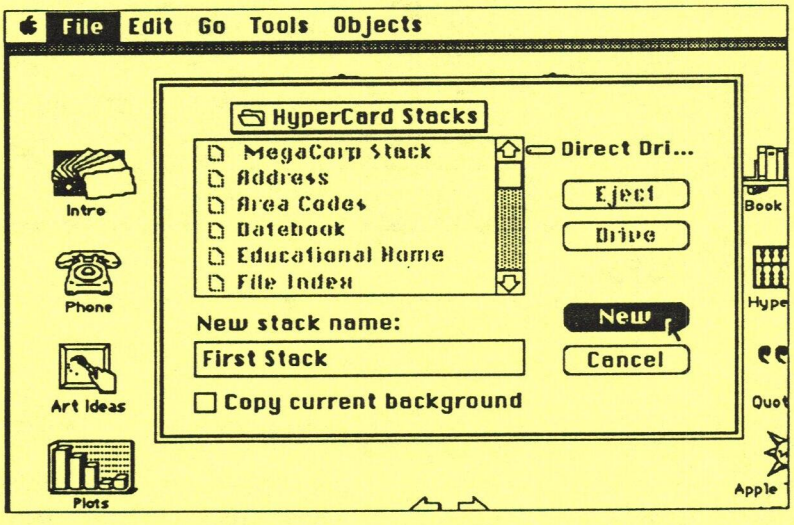


fig. 8d Click on the button marked New to create the stack. The stack is blank and is automatically saved on the current disk.

Do Not Copy the Current Background

Unselect the box that allows you to copy or change the background. This option gives us a chance to start with an entirely new stack or copy some of the characteristics of the stack you are already in. More important here is the notion of clicking on a box and changing a characteristic of the new stack. **(fig. 9)**

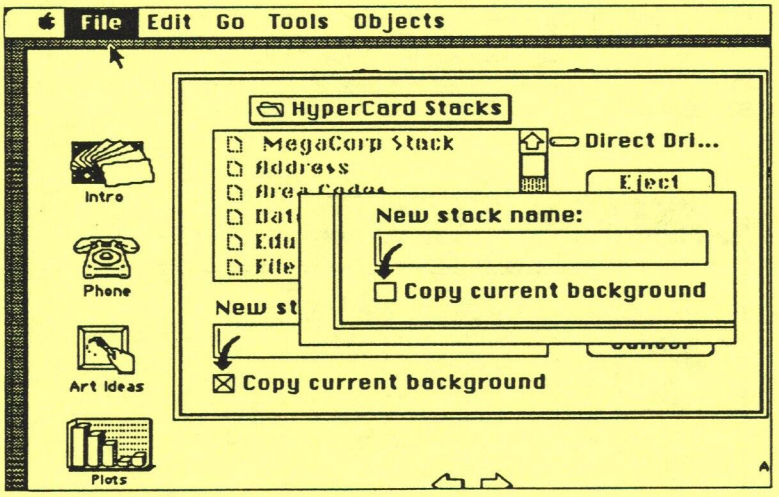


fig. 9 Practice clicking on a check box window to demonstrate ON and OFF.

Navigate	Dialog Box	Cursor	Save
Navigation is the movement around stacks by buttons or menu.	Interaction between the Mac and users occurs through questions asked in dialog boxes. Information about a new stack is necessary before the stack can be saved.	The on screen mouse changes forms often as its function changes.	HyperCard saves when the browse tool is selected.

Making the New Stack

We now have all the necessary ingredients for having the first stacks created by the participants. Click on **New** in the dialog box. After a few seconds there should be a blank screen with the HyperCard menus at the top. **(fig. 10)**

First Card is the Title Card

You should get into the habit of making the first card on the stack a title card. It's a little like remembering to write your name on a paper before handing it in. The title card should have the name of the stack, your name and the date.

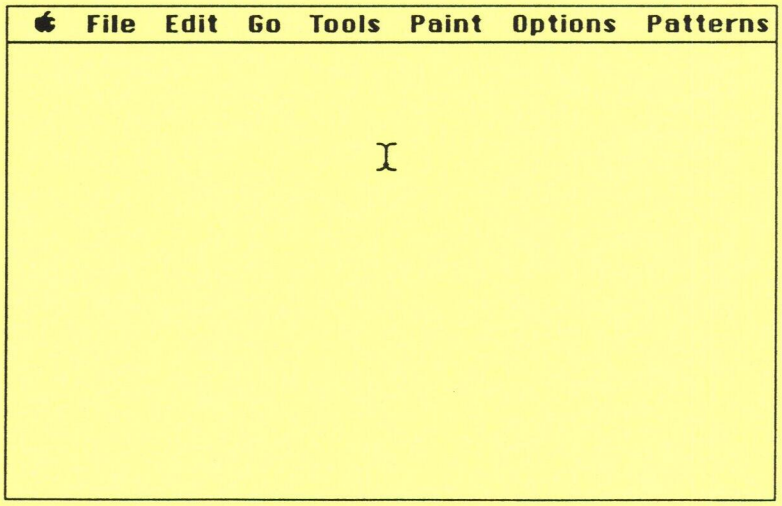


fig. 10 First Stack should begin with a blank screen . After naming the stack and unselecting the current background, click on New to create a new stack on the disk currently in use. HyperCard automatically creates the stack.

Background

There is a background on each card which can be designed separately from the transparent card layer. The background of a stack is decided when it's created.

Dialog Box

Interaction between the Mac and users occurs through questions asked in dialog boxes. Information about a new stack is necessary before the stack can be saved.

Toggle

Some buttons turn something either "on" or "off".

Creating A Simple Stack of Group Members

This stack is ultimate simplicity. It will be composed of a group of cards that equal in number the people in each row or convenient grouping. Each card will have information about a particular person in that row.

Save	Menus	Browse	Navigate	New Stack
HyperCard saves when the browse tool is selected.	The menus of HyperCard change with the tool selected.	The browse tool is used to click on buttons and fill in text.	Navigation can be done with menus, buttons and key strokes.	New stacks are named and the background set in a dialog box.

Demonstrate Filling in the First Card
Use Real Names in the Follow Me Stack

Since this is a Follow-Me, the leader will pick three names out of the room to include in the sample stack. You will then make your own stack with the people who are in your group.

Follow the leader to choose the text tool and then type on the first card in separate lines the **name, school, grade level, department and phone number. (fig. 11)**

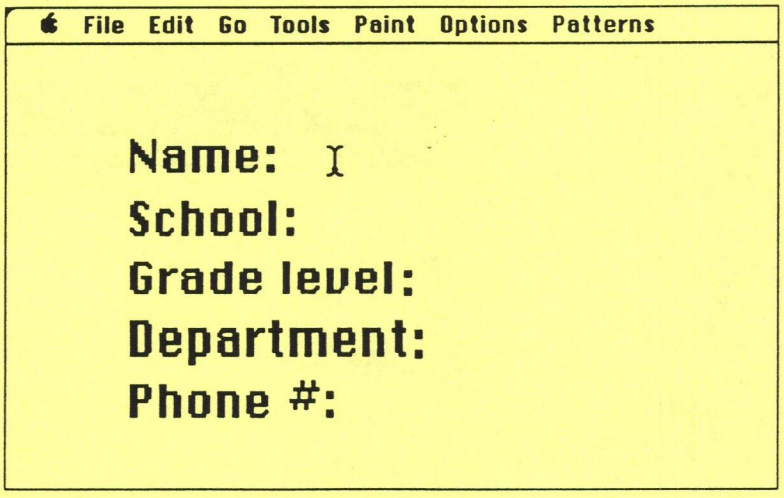


fig. 11 Each card on First Stack will contain information about a person in the group. Participants will take their time gathering information and filling in the card.

Text Tool	Background	Card	New Stack
The text tool is used to enter text in graphic form, not easily editted.	There is a background on each card which can be designed separately from the transparent card layer. The background of a stack is decided when it's created.	A card is one screenful of information, either text or graphic.	Every New Stack must have a name and background.

The Contents of the First Stack

You will have three or four cards per stack. You will fill in information about three other people and build the stack. Practice navigating around your stack each time you add a new card. **(fig. 12)**

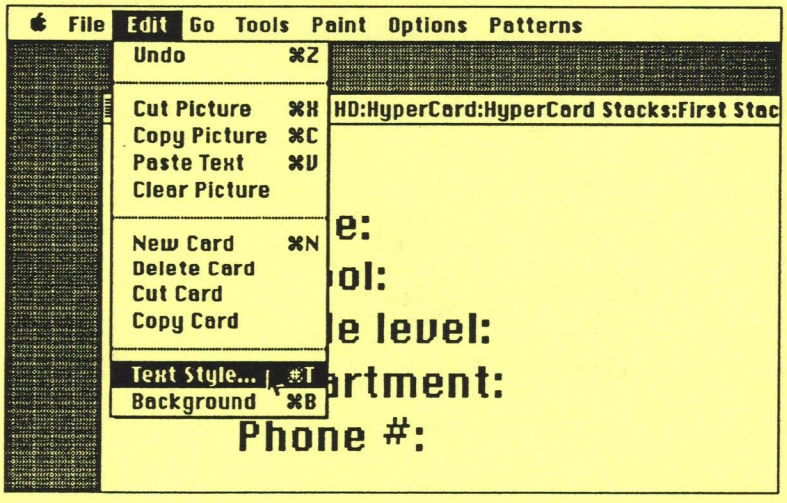


fig. 12 Information in First Stack will be filled in using the Text Tool. The font style, size and alignment of the text can be changed by choosing Text Style on the Edit menu.

Navigate	Menus	Browse	Save	Stacks
Navigation can be done with menus, buttons and key strokes.	The menus of HyperCard change with the tool selected.	The browse tool is used to click on buttons and fill in text.	HyperCard saves a stack when the browse tool is chosen.	Stacks are a set of cards used to save and organize information.

Allow time for 3 or 4 Cards to be Created Circulate to Answer Questions, Make Suggestions

When you get this first card finished, you are ready for the next step. The next step will be to get a new card set up just like this one, only with a different person from their group. **(fig. 13)** You may need to move around the room during this period to get the information you need.

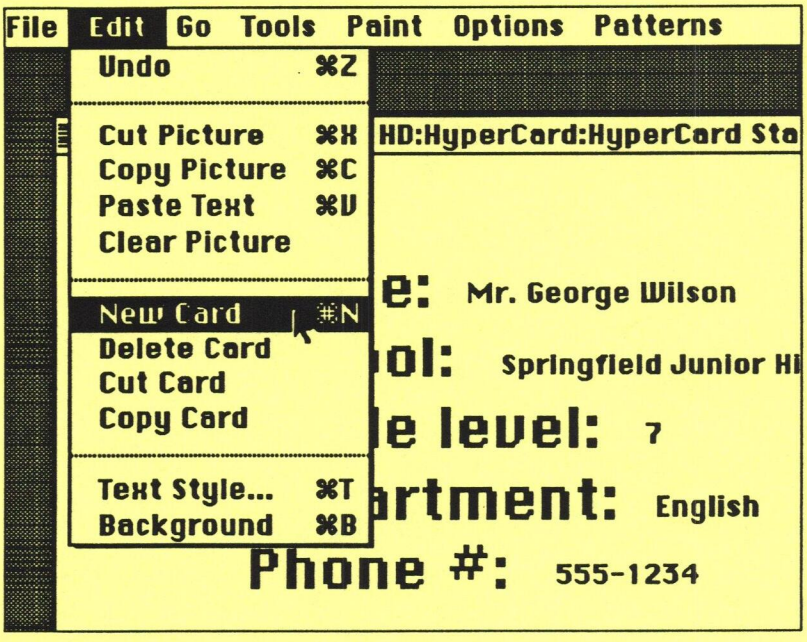


fig. 13 The New Card option on the Edit menu adds a new card to the stack. The new card is added in behind, following, the card you are looking at when you select New Card. Each New Card added to this stack will be blank.

Text Tool	HyperText	Mouse Click	Browse
The text tool is used to enter text in graphic form, not easily editted.	Non-linear, or hyper, text allows extensions of ideas to be invisibly attached to a document. A definition can be seen by clicking on the word.	The mouse click is an event which readies the Mac for some action.	The browse tool is used to enter text into a field.

Encourage Movement and Discussion

The information which you will gather for this stack is also information which you will find helpful during the course of the training program. As the number of educators using HyperCard increases you will want to be part of the network of users who can share experiences and ideas for applying HyperCard with students.

Edit	Background	New Card	Menus
The edit menu provides for card, button, field management.	There is a background on each card which can be designed separately from the transparent card layer. The background of a stack is decided when it is created.	When a new card is added to a stack it looks like the current card.	The menus of HyperCard change with the tool selected.

Break

Review and Preview

It is very important for you to have an opportunity to ask questions and talk about your progress. The leader will take a few minutes to review the concepts presented in the first session. Take this opportunity to ask about anything that is not clear before we go on.

Also at this time, preview what will be happening in the next session. You will create a stack of cards using the graphics or MacPaint tools. You will also practice navigating through your stacks using the Go menu. **(fig.14)**

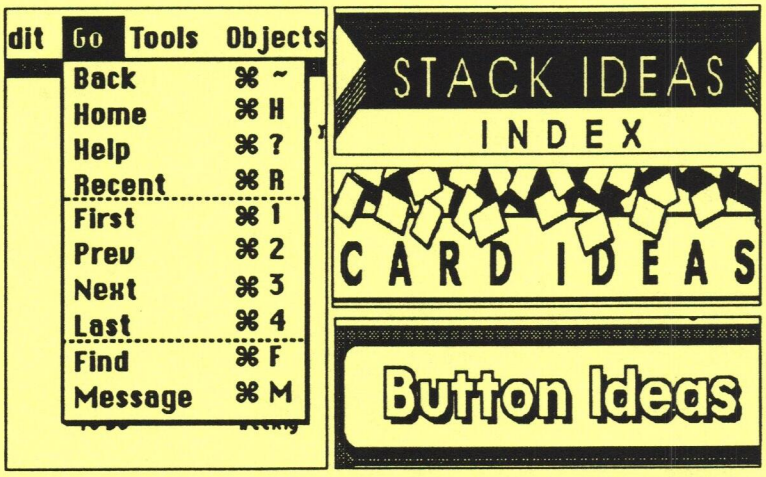


fig . 14 Review to reinforce the concepts presented in the first session. So far we have looked at stacks, cards and buttons. Time has been spent navigating among the objects of HyperCard.

Browse	Go Menu	Back	Wraparound
The browse tool is used to enter text into a field.	The Go menu offers many choices for navigation.	Back on the Go menu takes you "back" to the last card visited.	Navigating in a stack can be described as wraparound because choosing "Next" on the last card of the stack takes you to the first card of the stack.

Create New Cards as Review

Continue to work on the First Stack, adding new cards and filling in information about the people in your group. **(fig. 15)**

Notice again that the **New Card** option on the **Edit** menu adds a blank card to the First Stack. The New Card is added to the stack behind (following) the card that is presently visible. Try inserting cards in various locations in the stack to make this very clear. Practice cards may be deleted by selecting Delete Card on the Edit menu.

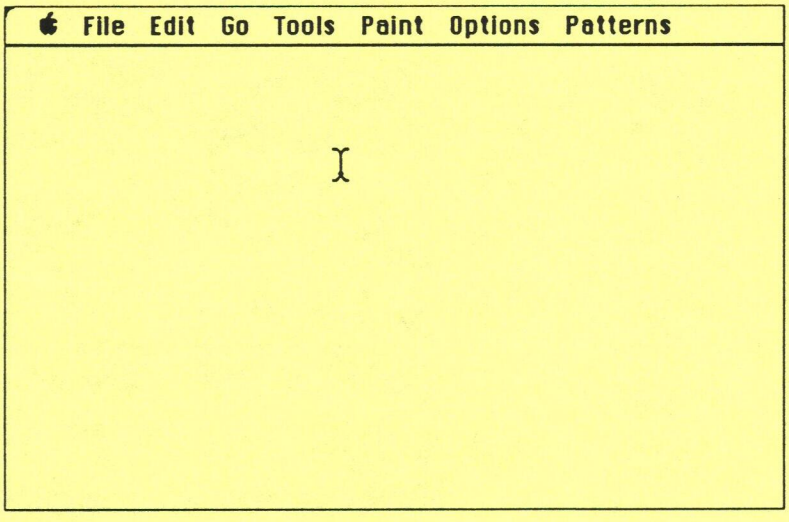


fig. 15 The Edit menu allows for New Cards to be added to a stack. New cards look exactly like the first card in the stack before any information is added.

Open Stack

The Open Stack choice on the File menu presents a dialog box listing the stacks on the current disk. The user may choose a stack or move to another disk directory.

Home

Home on the Go menu takes the user to the Home card.

Command Key

The command key may be used in place of the Go menu choices for navigation. Command 1=First, Command 2=Previous, Command 3=Next, Command 4=Last.

Practice Navigating

When everyone has completed the several cards do some practice navigation exercises in the Follow-Me mode. You will use the **Go** menu choices: **Next, Previous, First,** and **Last.** Notice that some of these are duplicates. (First gives the same result as Previous if you are on Card 2.) Furthermore notice the "wraparound" effect if you choose **Next** successively. **(fig. 16)**

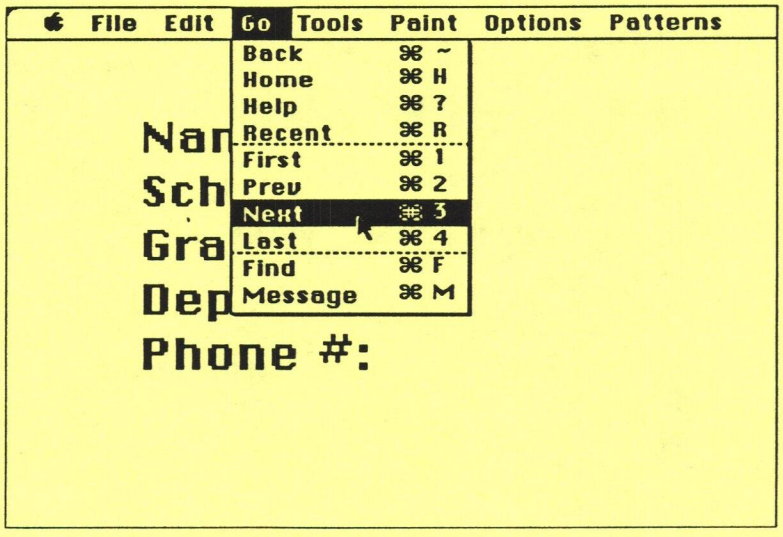


fig. 16 Go menu provides several ways to navigate around a stack. First, Prev, Next, and Last are most commonly used.

Open Stack

The Open Stack choice on the File menu presents a dialog box listing the stacks on the current disk. The user may choose a stack or move to another disk directory.

HyperTalk

Programs written to control the objects of HyperCard are written in a programming language called HyperTalk. Programs in HyperTalk are called scripts.

Save

Saving occurs automatically when the browse tool is selected .

Returning Home
Opening a Stack using the File Menu

After some practice go back to the **Home Card** by picking **Home** from the **Go** menu. We have navigated back to the Home card, the starting point by the mere click of the button. HyperCard makes such smooth connections that we can use them without remembering the electronic pathway.

Now you will learn how to use **Open Stack** on the **File** menu to get back to your Stack. **(fig. 17)**

Try these progressions:

1. Quit HyperCard, find the First Stack on the desktop, open the First Stack.

2. Quit HyperCard,open the Home stack, use Open Stack (File menu) to open First Stack.

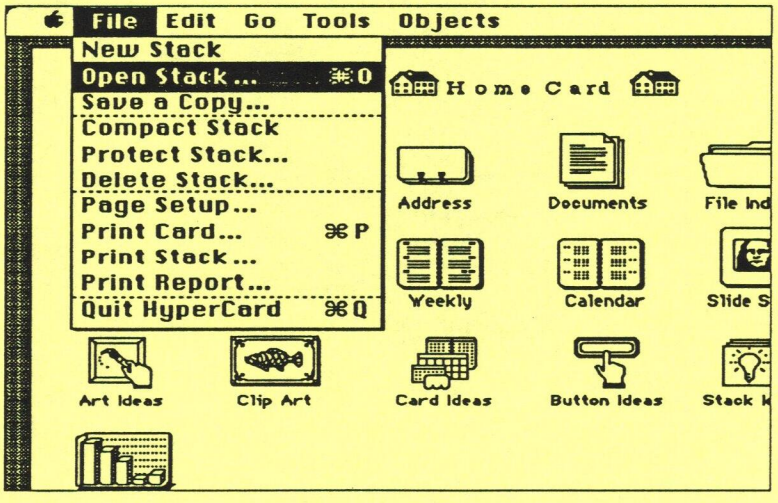


fig. 17 Open Stack on the File menu allows for navigation between stacks when buttons are not available.

Save

HyperCard saves automatically when the browse tool is selected.

Menus

The tools and patterns menus can be moved by clicking on the top bar of the menu and dragging the menu to a new loaction on the screen.

Tear-off Menu

The tools and patterns menus may be "torn off" the menu bar by dragging the mouse down and off the menu.

HyperCard and the Automatic Save

A word should be said here about the automatic Save feature of HyperCard so that you understand this. The automatic save happens when you return to the browse tool. If you are in the middle of your rendering of the Madonna with Child when the power goes off, you lose everything since your last return to the browse tool.

Complete the First Stack

You may now go back to the First Stack and complete your cards on your own. This will take perhaps 10 minutes or so.

Open Stack

The Open Stack choice on the File menu presents a dialog box listing the stacks on the current disk. The user may choose a stack or move to another disk directory.

HyperTalk

Programs written to control the objects of HyperCard are written in a programming language called HyperTalk. Programs in HyperTalk are called scripts.

Save

Saving occurs when the browse tool is selected or Keep is chosen.

Tear-off Menus

When you have finished the three or four cards on your stack, go back to the first card on your stack for one final Follow Me exercise. The leader will demonstrate how to tear off the tool palette and drag it down to the lower right or left hand corner of the screen. Try it out for yourself and then click on the top of the menu bar and move it around the screen. **(fig. 18)**

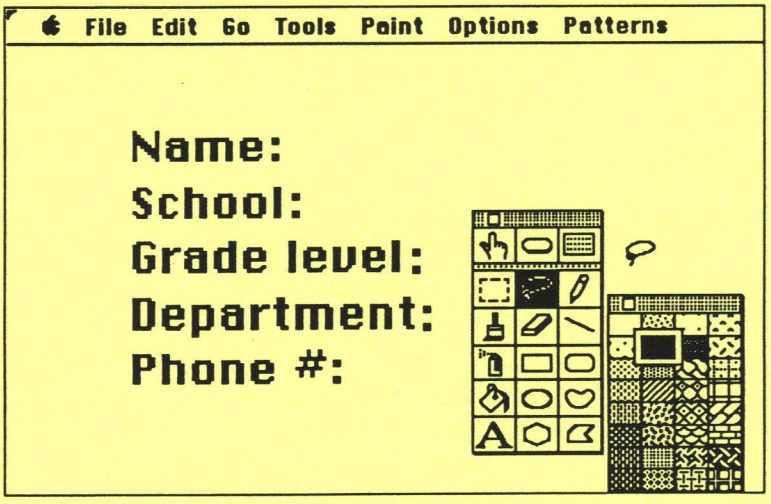


fig. 18 Tear off menus can be moved around the screen by clicking on the bars at the top and dragged to a new location. The small box in the upper left corner closes the menu.

Tear-off Menu	Menus	Save
The tools and patterns menus may be "torn off" the menu bar by dragging the mouse down and off the menu.	The tools and patterns menus can be moved by clicking on the top bar of the menu and dragging the menu to a new loaction on the screen.	HyperCard saves automatically when the browse tool is selected.

A Final Stack as Review

There may be time for one more short project before the end of the session. You will make one more **New Stack** from the File menu. Don't copy the current background. Name the stack Navigation Stack. Fill out the title card with the name of the stack, your name and the date.

The stack will have three cards, each with part of a complete picture. For example, the first card might have the bottom of a snowman, the second card would have the bottom and middle of the snowman and the third card would have the entire snowman. **(fig. 19)**

Once the stack is completed review navigation by moving around the stack using the Go menu or the keyboard strokes. You may add more cards in order to practice the means of navigation.

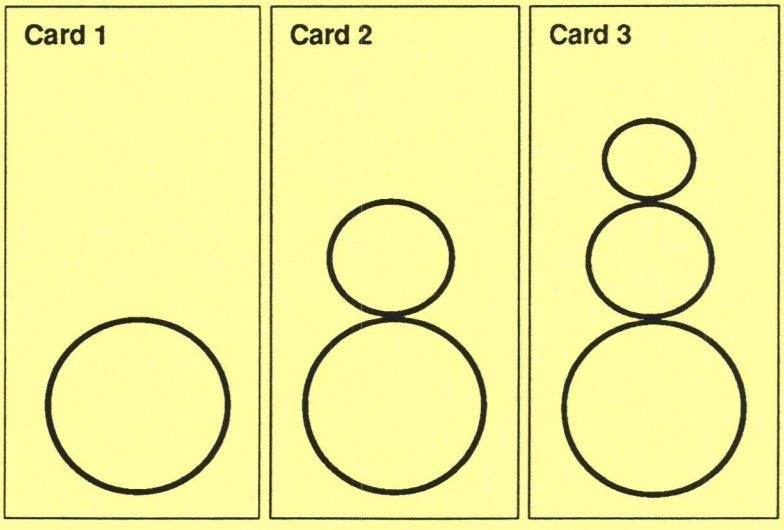


Fig. 19 The short Snowman Stack will reinforce the idea of stack building and navigation. The first card has only one part of the picture. The second card has two parts and the third card has the complete picture. Navigate around the stack using the Go menu or the keyboard strokes to see the picture.

Bill Atkinson

Bill Atkinson is retained by Apple Computer Inc. as an Apple Fellow to help provide a vision of the future. He is the author of MacPaint and HyperCard.

References

Reference books for HyperCard include Danny Goodman's *The Complete HyperCard Handbook*, Dan Shafer's *HyperTalk Programming*.

Quit

A work session ends when Quit is chosen on the File menu.

Conclude the session with Review and Evaluation

We conclude the module with a review of the skills, vocabulary and concepts of HyperCard covered in this module. You should be able to **identify cards, stacks, and means of navigation** at the end of this module.

Everyone should write some comment on the grafitti evaluation board at the end of this module.

Tear-off Menu

The tools and patterns menus may be "torn off" the menu bar by dragging the mouse down and off the menu.

Menus

The tools and patterns menus can be moved by clicking on the top bar of the menu and dragging the menu to a new loaction on the screen.

Save

HyperCard saves automatically when the browse tool is selected.

HyperCard in Education

Module 2 Participant's Guide

Joseph F. Hofmeister • Joyce B. Rudowski
The HyperSchool
Cincinnati Country Day School
6905 Given Road, Cincinnati, Ohio 45243
513-561-7298 AppleLink K0197

Module 2

Introduction - Fielding Questions

Session two begins with a short review, question and response period. Question and response here, as opposed to question and answer, is deliberate. You will realize as you use HyperCard that "the answers" are not simply "out there" somewhere for you to just tap into a carefully defined and definitive textbook whenever there is a problem. HyperCard actually frees us from that model of teaching. That's a model where the teacher knows all the answers and the work done in schools boils down to getting these answers into the heads of the students. HyperCard by its nature encourages a model of teaching where the teacher and students find their way to knowledge together.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Answers Discovered Together

Questions that come up in this period will be welcomed as a chance for the group to bounce ideas back and forth. Take this opporuntity to ask about anything that confused or amazed you during the first session.

Think about the skills you learned and how you might use them in the classroom. What kind of assignments might you make that would include HyperCard solutions? What more do you want to know about building a stack and navigating in it?

Button	**Field**	**Import**	**Script**	**Save**
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Review New Skills

The skills that you should have acquired during the first session are:

Stacks, with different items on individual cards in the stack

Navigating to different cards in the stack by use of the **Go** menu

The circular nature of a stack

How do you get to the beginning of that stack? How do you get **Home?**

Remember that the keyboard equivalents of **Previous** and **Next** are **command-2** and **command-3**. The keyboard equivalents are on the menus in case you forget what they are. **(fig. 1)**

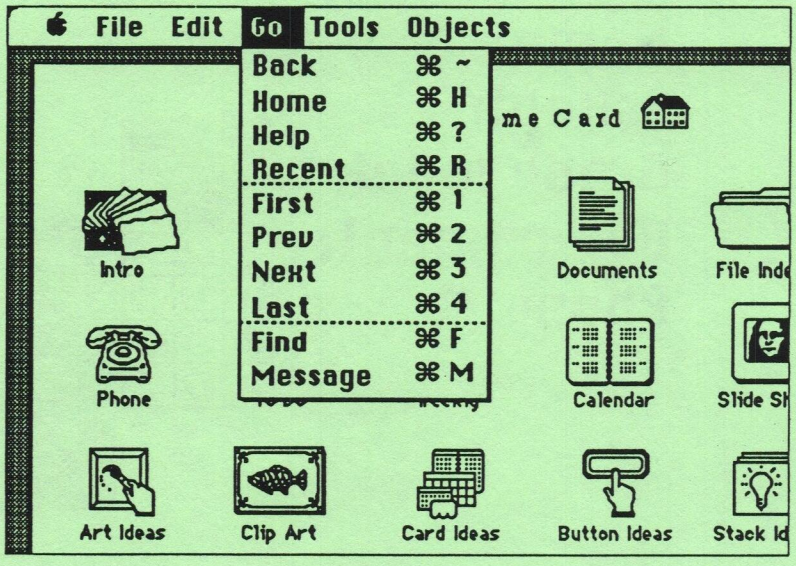


fig. 1 The Go menu provides immediate navigation by selecting Next, Prev, or First, etc. This menu also shows the keyboard equivalents for the menu choices.

Stack	Card	Navigation	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Navigation includes a variety of ways of moving from one card to another. Navigation may be done by menu, keyboard or buttons.	The Go Menu is the primary means of stack navigation.

Review the Basics

Remember that the menus tear off and may be moved around the screen. Further, the Text tool is used to type information on a card. This as a good place to begin this module as we will learn another way to handle text in a stack. **(fig. 2)**

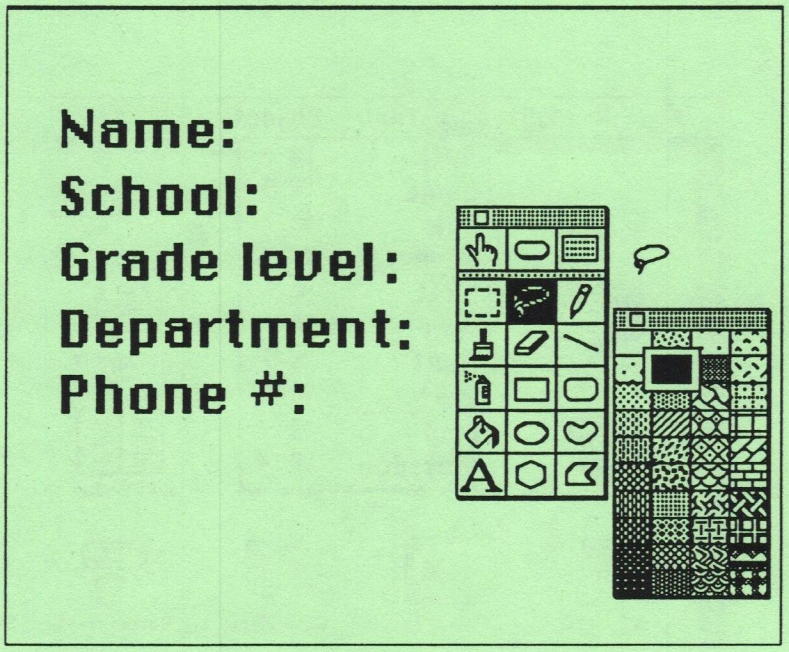


fig. 2 The Tool and Pattern menus "tear off" of the menu bar and may be repositioned on the screen.

Button	Field	Tear Off	Tools	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	The tools and patterns menus "tear off" and relocate.	The tools allow for the creation of objects and graphics.	HyperCard saves automatically when the browse tool is selected.

Making Text Fields

We will now do a Follow Me exercise that creates a new stack. Get to the **Home** card by double clicking on the Home stack or HyperCard. Then select **New Stack** on the **File** menu. Name this stack **Field Stack** and **toggle out of** (turn off) the Copy Current Background option. **(fig 3a-b)**

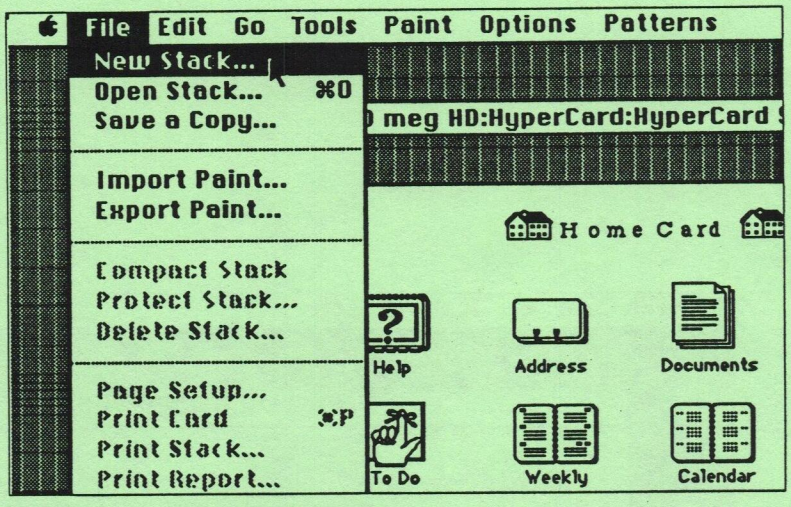


fig. 3a The File menu contains the choices pertinent to stack creation and management. Select New Stack to get the dialog box which creates a new stack.

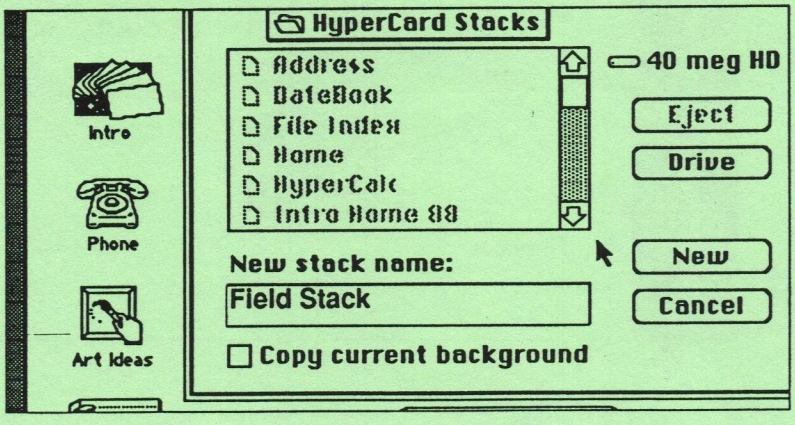


fig. 3b The New stack dialog box asks you to name the stack and decide whether or not to copy the background of the current stack.

Stack	New Stack	Background	Go Menu
A stack is the file of information created by HyperCard.	A new stack is named and the background copied or not.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Identifying the Field Tool

When you are on the first card of the new stack, tear off the tools menu and put it in the lower right hand corner. Now notice the three major HyperCard tools, the **Browse, Button** and **Field** modes across the top of the tear off tools menu. **(fig. 4)**

These three tools are very different from the other choices on this menu. Thus far we have been using only the browse tool.

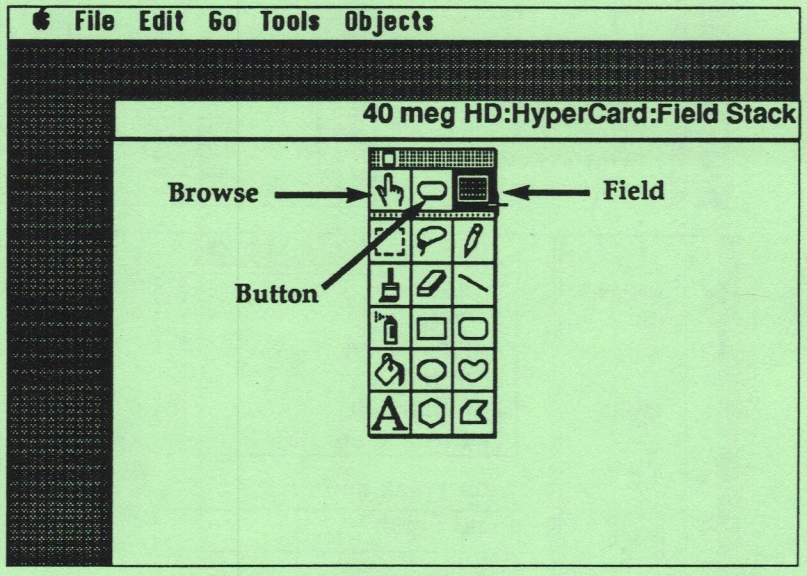


fig. 4 The Tear-off Tools Palette contains the basic HyperCard tools to browse through a stack as well as create buttons and fields.

Button	Field	Import	Script	Field Tool
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	The field tool on the tool menu manages the design of a field.

Making a New Field

Follow the leader to choose the field tool, and then pick the **New Field** choice off the **Objects** menu. **(fig. 5)**

Make several new fields on this card. Resize the fields by clicking and dragging on the corner of the fields. Move the fields around by clicking on the middle of the field and dragging it to a new place. Now choose the browse tool again and don't panic when the fields disappear from view.

Choose the field tool and click on one of the fields. Now get **Field Info** from the **Objects** menu and change the style of the field so that it can be seen.

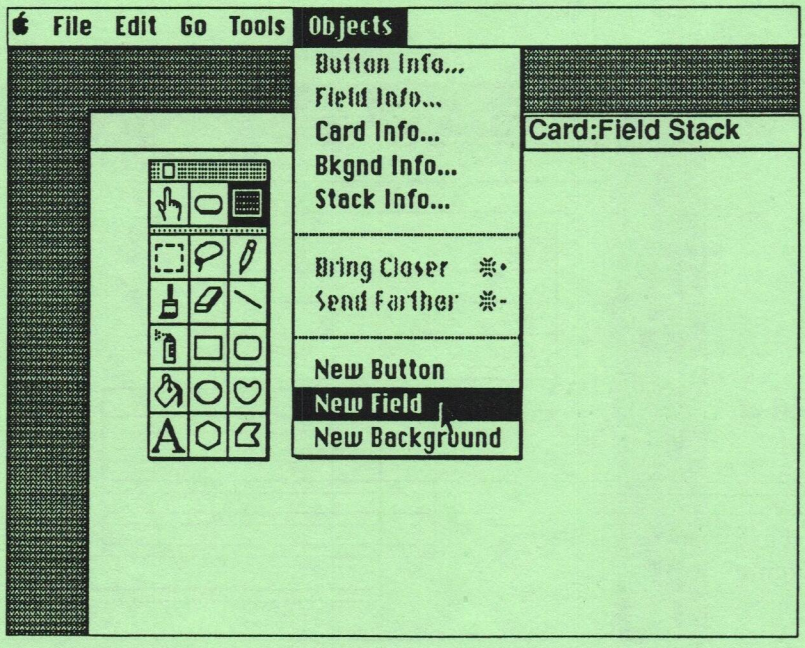


fig. 5 The Objects Menu provides choices which control the creation and manipulation of the objects of HyperCard. Objects have names and numbers for specific identification. It is important to be able to identify the objects of a stack.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Duplicating Fields and Changing Fonts

You can easily duplicate and keep fields orthogonal. This is done by holding down the shift and option key at the same time that you click on the field and drag a copy on the card. **(fig. 6)**

Follow the leader to make five fields on the screen and enter one word in each field. You can change the font of the field by choosing **Field Info** on the **Objects** menu and then choosing **Font** on the Field Info dialog box. Change the font style for each of the fields you have created.

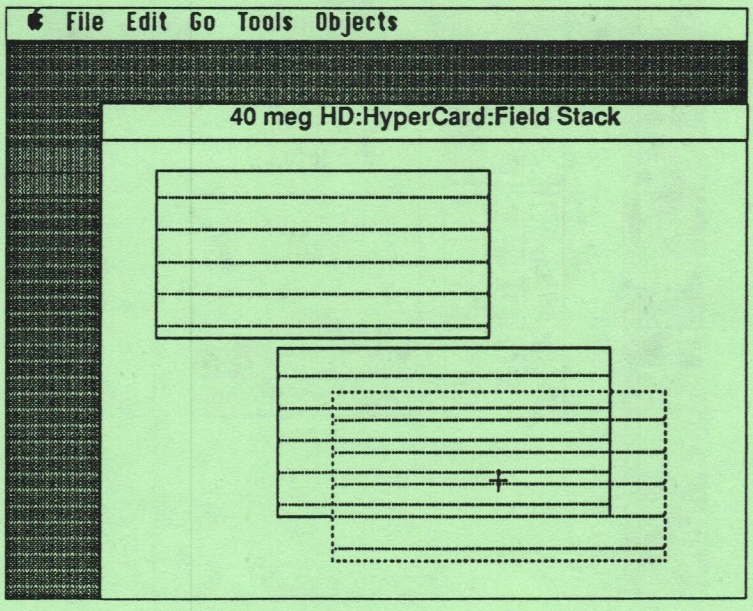


fig.6 The Option key, on the lower left of the keyboard, provides the function of duplicating objects on the screen. Both fields and buttons can be duplicated by creating one and then holding down the option key while clicking on and dragging the field or button on the screen.

Button	Field	Option	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	The option key can be used to make duplicates of fields.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Entering Information into a Field

The biggest problem most beginners have with text fields is entering information into them. The browse tool must be used to type into a field.

Field Info Identifies a Field

A field is active when the edges are "wiggling". Only an active field can be changed, moved or re-sized. Make one of your fields active and then go to the **Objects** menu and find **Field Info**. Select **Field Info** for the field and notice the various options available, especially the Transparent and Show Lines choices. **(fig. 8)**

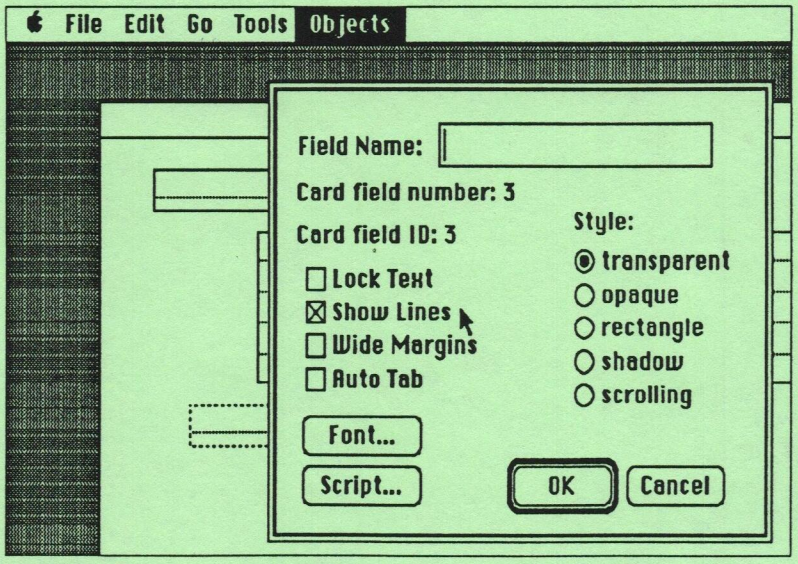


fig. 8 Field Info contains choices to custom design a field. The style of the field can be determined as well as the text style and font size. Notice that the lines in the field may be shown or not shown.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Build A Stack with A Background

We will now create another new stack. This one will be called **Tree Stack**. Go to **New Stack** on the File menu, name the stack **Tree Stack** and click off the current background. **(fig. 9)**

We will work with both **Background** and card levels in this stack, introducing the notion of background as another layer of HyperCard.

We get to the background by choosing **Background** on the **Edit** menu or **command-B** on the keyboard.

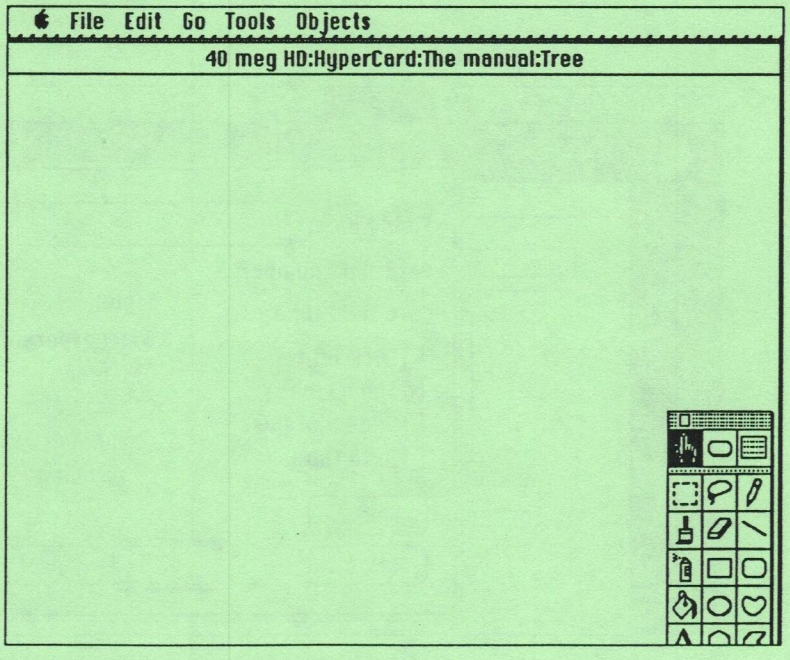


fig. 9 A new stack called Tree Stack will for the first time use the background as an independent part of the stack. A tree will be drawn freehand on the background of the stack.

Button	Field	Field Info	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Field info sets the name, style and font of a text field.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Dealing with the Background

This will be confusing at first, since this is the first actual experience you have had in dealing with backgrounds. Watch as the leader moves to the background on the big screen and notice the **hash** marks across the menu bar when you are in the background.

Move to the background of your stack and draw a tree using the paint tools. The tree should have no leaves on it. **(fig. 10)**

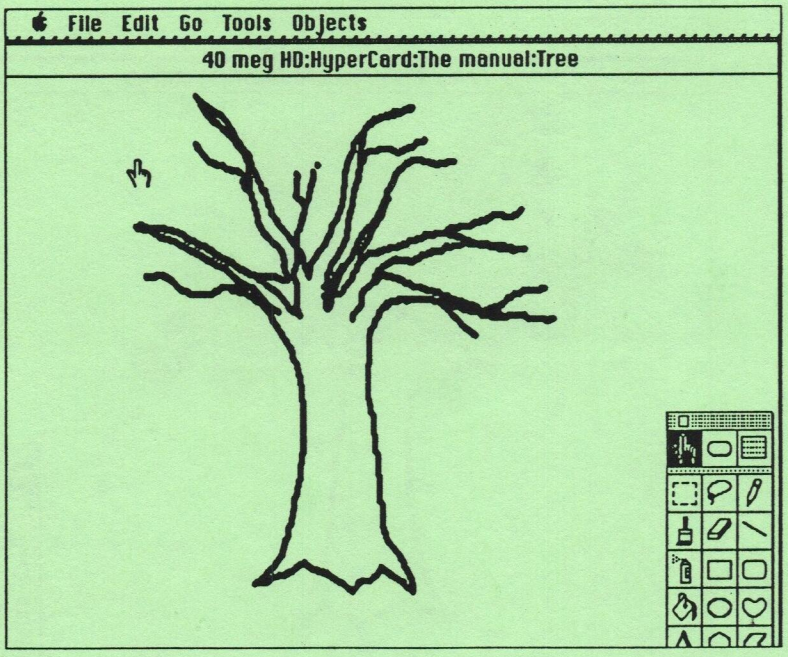


fig. 10 Draw a tree with the paint tools on the background of the stack. The background is a separate level of each card.

Stack	Cursor	Background	Go Menu
A stack is the file of information created by HyperCard.	The browse cursor changes to an I-beam on a field.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Draw Leaves on the Card Level

When the bare tree is finished return to the card level by again choosing **Background** on the **Edit** menu. You may want to repeat this process two or three times until the idea of two levels sinks in.

Now draw leaves on the tree on the card level. When this a complete, you will see what looks like a complete tree in leaf. Now choose **Background** again and see that the bare tree is still on the background. Move back and forth several times. **(fig. 11)**

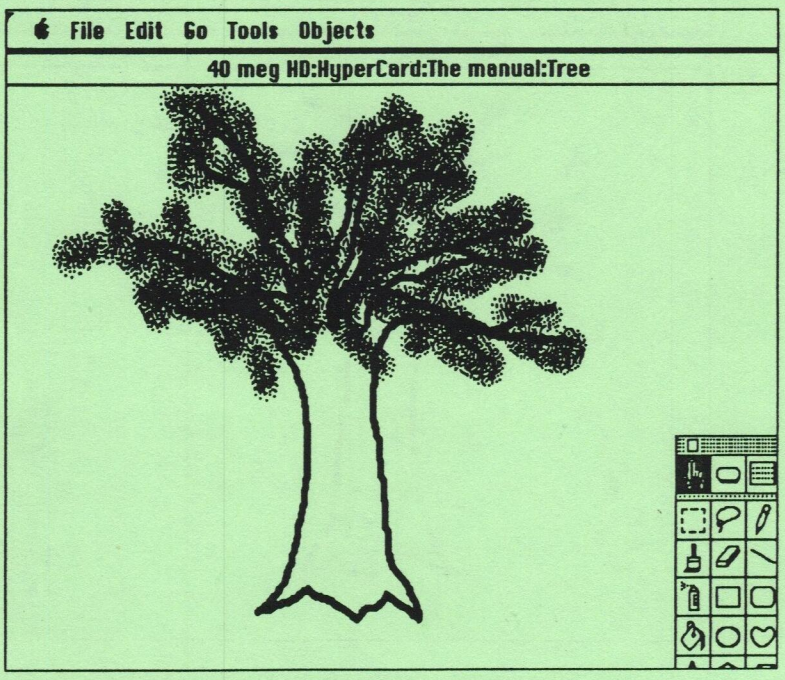


fig. 11 Add leaves to the picture on the card level. Now go back and forth between the card level and the background to confirm the fact that the tree exists only on the background and the leaves exist only on the card level.

Button	Field	Delete	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	A stack may be deleted by choosing Delete Stack on File.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Add Another Object to the Background

Now you will add another object to the background of the stack. Check to be sure that you are on the background before you begin to draw. The added object might be a bird flying toward that tree or a dog sitting under the tree. The addition should not cover any part of the tree however.

When the drawing is complete move to the card level and confirm that the object is still visible on the background. **(fig. 12)**

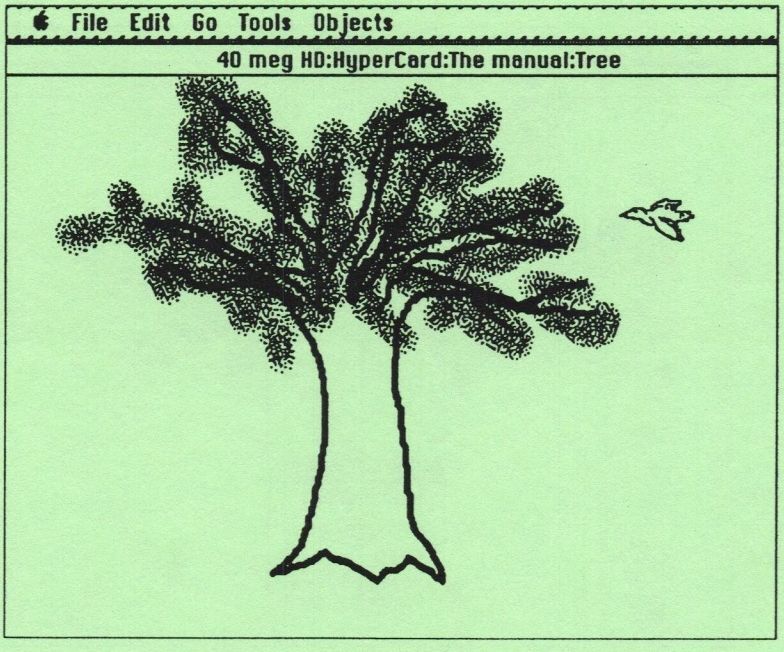


fig. 12 The tree is now joined by a bird flying its way in the background of the stack.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Use Opaque to Block Out Part of the Background

On the card level of the stack, draw the dotted selection box around the area where the bird or extra object is on the background. This selects that area of the screen. Now go to the **Paint** menu and choose the **Opaque** option.

The object on the background "disappears" when the card level is made opaque or "painted in". Now select the same area and choose **Transparent** from the **Paint** menu. Repeat this process several times.

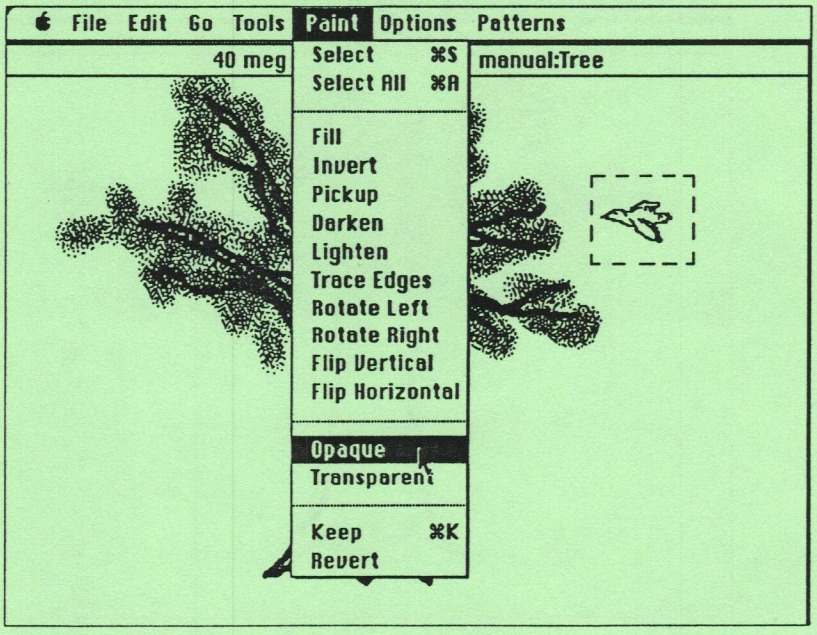


fig. 13 Use the Select tool, the dotted rectangle, to surround the extra object on the picture. The Opaque choice on the Paint menu allows this section of the card level to block out the section of a background.

Button	Field	Import	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Break

Review and Digest

After the break we will review and digest the material covered in the first session.
Recall the process of making and changing a text field, including duplicating a field and changing the style and font.

We will also review the notion of background and card level stressing the identification of the background through the hash marks on the menu bar. Think about the relationship between background and card level, and the ability to hide parts of the background using opaque.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Disciplinary Stack

We will now begin a new stack using the skills learned in the first part of this module. This is a "talk through" with the trainers using a blackboard or poster board to demonstrate what is involved with this project. The stack will be named **Disciplinary Stack.**

You will now make a stack composed of **five** cards, (title card plus four more) all having the same background. The background will be four words or pictures across the top of the page. In mathematics, for example, the four words written on the background might be four types of quadrilaterals: trapezoid, parallelogram, square and rectangle. These words are to be typed equidistanced across the top of the background. **(fig. 14)**

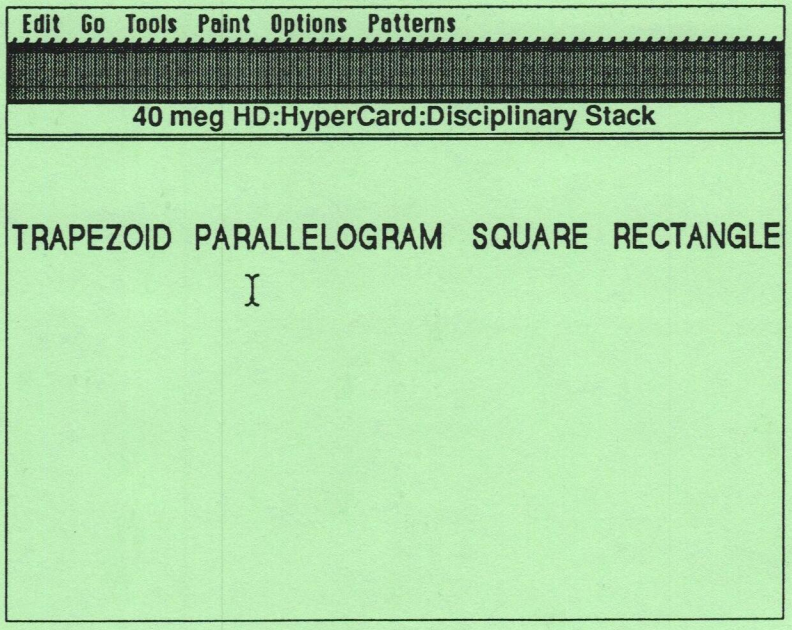


fig. 14 Four words from your discipline will be put on the background of this stack. You may want to draw pictures as an alternative.

Button	Field	Import	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Four Names and Four Fields

The goal of this exercise is to have the four names or pictures at the top of each card and then to have a field directly under each word or picture, explaining it. The four names or pictures are to be on the background. There will be a field on each card on the card level.

On each of the next four cards, the user should see only one word (from the background) and the explanation in the field below it (from the card level). **(fig. 15)**

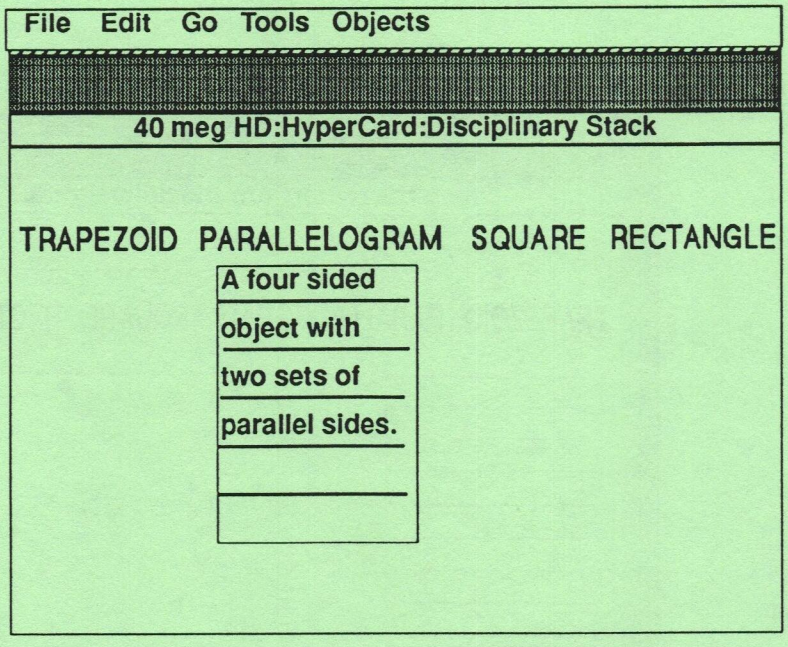


fig. 15 After the title card there will be a card with four words and four fields for explanations or definitions of the words.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Four Cards with Only One Word and Field

You will begin by making four cards in addition to the title card. Each time you choose **New Card** on the Edit menu a card will be added which looks exactly like the original card.

On each of these four cards only one of the words will be seen and one field with the definition or explanation of the word. The other three words will be opaqued out using the dotted rectangle tool, or **Selection Box**, (right below the browse tool on the tool palette) and **Opaque** under the **Paint** menu. **(fig. 16)**

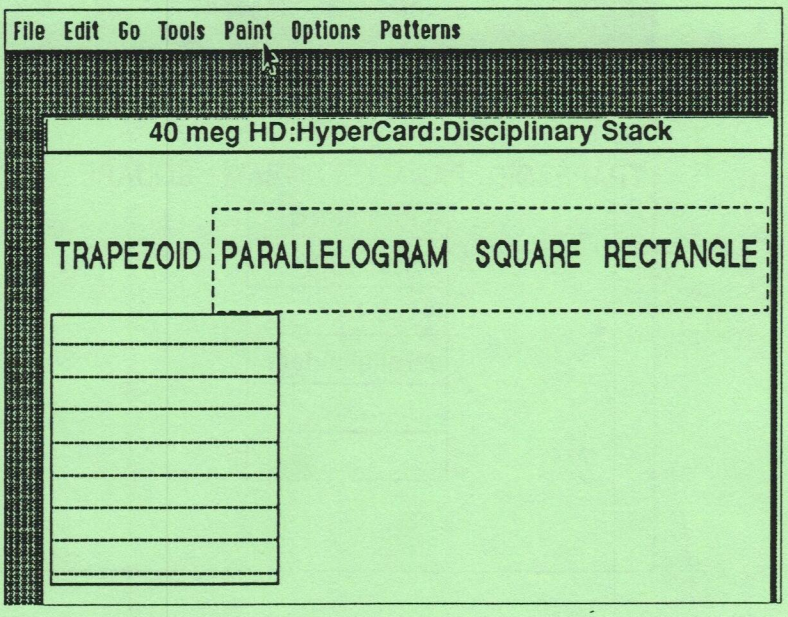


fig. 16 One card will have all four words. Each of the next four cards will have only one word and definition. The other words will be opaqued using the option **Opaque** on the Paint menu.

Lasso	Field	Copy	Script	Clipboard
The lasso tool draws an elastic line around a graphic item.	A field adds the capabilities of a word processor to HyperCard.	Copy puts a "copy" of the selected item on the clipboard.	A script is the programming code which directs objects.	Clipboard is a temporary holder which contains 1 item at a time.

Try Out Opaque and Transparent

Take your time to practice with this exercise. Try out **Opaque** and **Transparent** several times until they are comfortable with it.

Choose **Opaque** from the **Paint** Menu and see that you can hide one of the words. Choose **Transparent** from the **Paint** Menu now and notice that the picture reappears.

Next add a field to each card and fill in the definition or explanation.

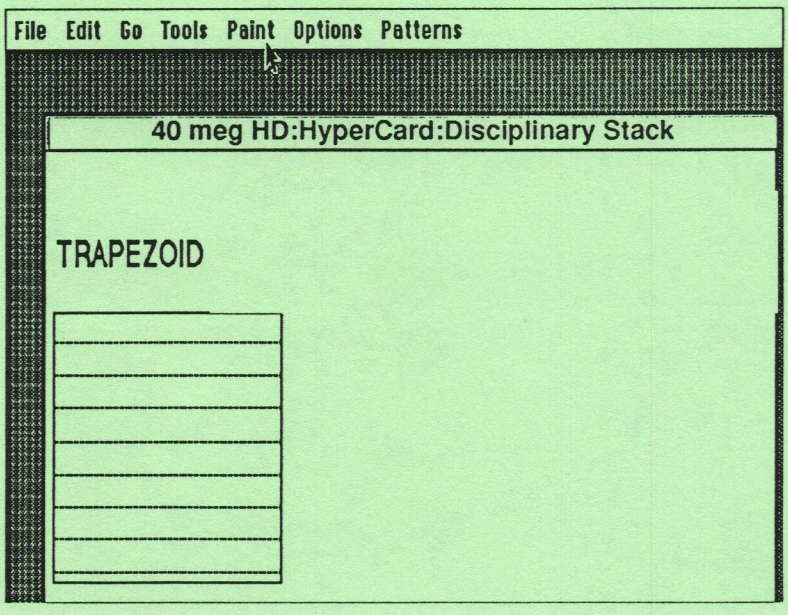


fig. 17 Each of the next four cards will have only one of the words and one text field on it.

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Time to Make a Stack

Once the process is understood, you will want to take some time to navigate around the stack. Finish the fields on each card and fill in the information for each word. Once again navigate around the stack using either the Go menu or the keyboard equivalents.

Button	Field	Opaque	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Opaque makes the selected area of the screen solid white.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Suggestions for the Stack

Be sure that you start with a title card (it's so much easier to put it in first then try to rearrange later). Title card should include your names, school and subject area, the date, and the title of this stack. You can use larger than normal type with the text tool, but start low enough on the card so that these things do not interfere with the items which will be put on the background on card two and then opaqued out on this card.

Summary of requirements for this stack:

1. **Minimum of 5 cards, counting one title card.**

2. **Four pictures (or four words with text tool) on the background of the stack, with the parts not desired on a particular card opaqued out.**

3. **Fields on the card level underneath each picture or word with some explanation or definition.**

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Completing the Stack

This exercise may take 10 or 15 minutes. Your completed stack will be a set of electronic flash cards. In this case the students could prepare their own stack of cards to review the parts of speech, types of quadrilaterals or French verbs.

Button	Field	Import	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

Conclusion
Review and Responses

Congratulations are in order here for having learned the five basic building blocks of HyperCard. Stacks, cards, backgrounds, fields and buttons have all been worked with during these first two modules.

Point out that there are five basic **elements**, or **objects**, in HyperCard. These five objects or elements are:

1. **Stacks**
2. **Backgrounds**
3. **Cards**
4. **Fields**
5. **Buttons**

Stack	Card	Background	Go Menu
A stack is the file of information created by HyperCard.	A card is a screenful of information, both text and graphic.	Every card has background which may contain text and graphics separate from the card level. There may be more than one background per stack.	The Go Menu is the primary means of stack navigation.

Graffiti Evaluation

You should fill in the graffiti evaluation board before you leave. The graffiti evaluation as mentioned in Module 1 provides a candid forum for responses to HyperCard and its uses.

Write one or two statements or phrases which might respond to these thoughts: "What was the one most important new skill you learned in the session", or "I discovered that I learn best when...."

Button	Field	Import	Script	Save
A button is a programmable object, often used for navigation.	A field adds the capabilities of a word processor to HyperCard.	Import is a menu choice providing for graphics to be used in a stack.	A script is the programming code which directs objects.	HyperCard saves automatically when the browse tool is selected.

HyperCard in Education

Module 3 Participant's Guide

Joseph F. Hofmeister • Joyce B. Rudowski
The HyperSchool
Cincinnati Country Day School
6905 Given Road, Cincinnati, Ohio 45243
513-561-7298 AppleLink K0197

Module 3

Introduction
Review and Preview

We start this module by taking a good look at what we accomplished in the first two modules. Since you have completed and understood the first two modules you have made excellent progress. Have you had a difficult time explaining exactly what you're doing? You may begin to appreciate the fact that answering the question "What is HyperCard?" isn't easy.

We will review by looking at the Disciplinary stacks from Module 2. Recall the basic skills of stack building, adding cards and using the paint tools. You may be concerned about educational application of what you have done. Think about the long range goal of empowering students to use HyperCard as a learning tool, rather than HyperCard as a teacher presentation tool.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designedobject of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Review Navigation

A nice anchoring point to start with in this review is the concept of navigation. So far you have learned how to navigate using the Go menu and perhaps the keyboard command equivalents.

We will introduce the button method now. This is a follow-me exercise, using the Disciplinary stacks for practice. Navigate through the stack using the menu and the keyboard commands.

Now tear off the Tools menu and select the button tool which is between the browse tool and the field tool on the top of the menu. **(fig. 1)**

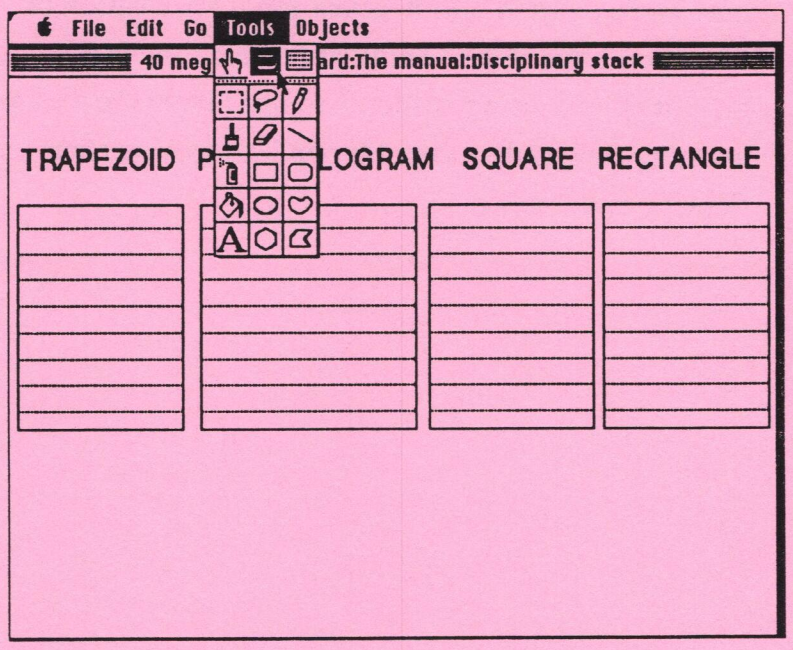


fig. 1 The Button tool, next to the browse tool, allows the creation and manipulation of buttons. Buttons may be placed on the background or card level. Buttons on the background are called background buttons. Buttons on the card level are called card buttons.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Buttons are created from the Objects Menu

Now choose **New Button** from the **Objects** menu and notice that the new button appears on the title page. Practice changing its shape and its location on the card by clicking on the corner to change the shape or in the middle to move it. Don't worry about the "New Button" title appearing on it for the moment. We want this first button to be a navigational button that takes the user from the title page to the first card.

Finally locate the button in the lower right hand corner of the card, where "next card" buttons are customarily placed. **(fig. 2)**

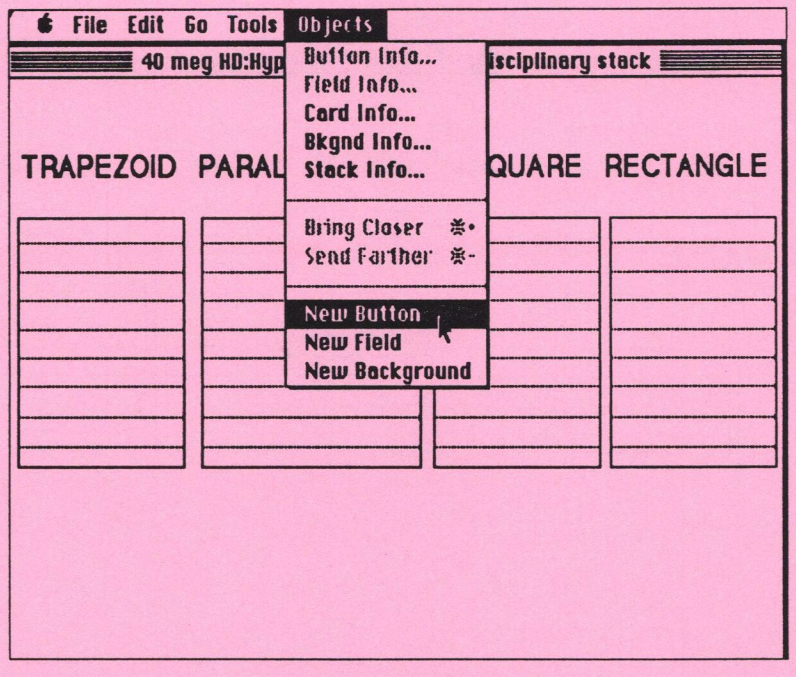


Fig. 2 The New Button choice, on the Objects Menu places a new button on the card.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designedobject of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Changing the Style of the Button

Now your button should be in the lower right corner, and it should be about 3/4 inches square. Somewhat larger is all right, but very small wouldn't be good, because we want to fit the customary right arrow on this button. You will notice that a button is sized and moved in much the same way as a field. **(fig. 3)**

Now go to **Button Info** on the **Objects** menu to get the dialog box for this button. Notice the different options here for buttons.

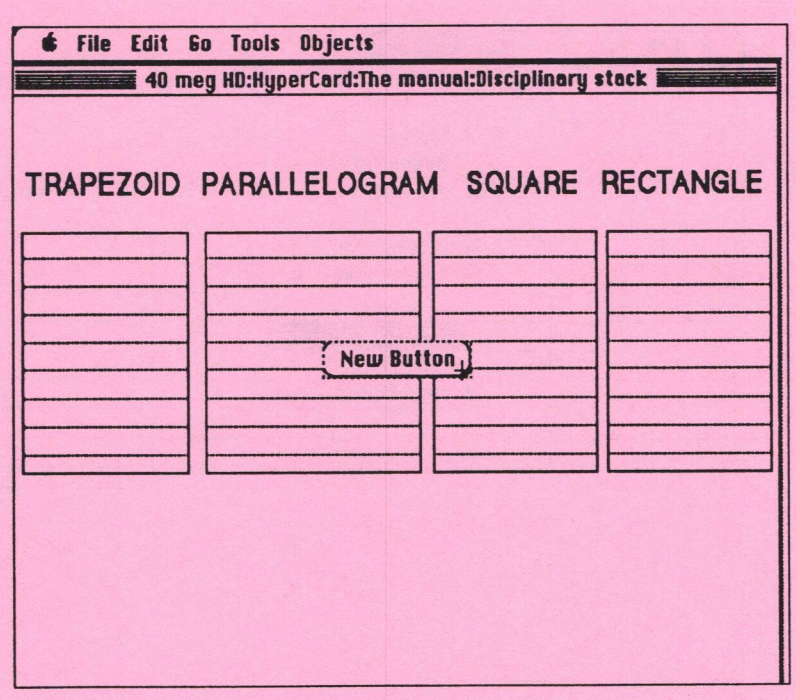


fig. 3 The New Button will be used to navigate from one card to the next. Ordinarily such a button is placed near the bottom of the card in the corner. Notice that buttons are handled much the same way as fields.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Changing the Style of a Button

Point out the similarities of the Field Info dialog box and the Button Info dialog box.

Untoggle, unselect, the **Show Name** option so that the "New Button" name on the button will not show on the card. Choose **transparent** so that the button will stand alone when the button is complete. **(fig. 4)**

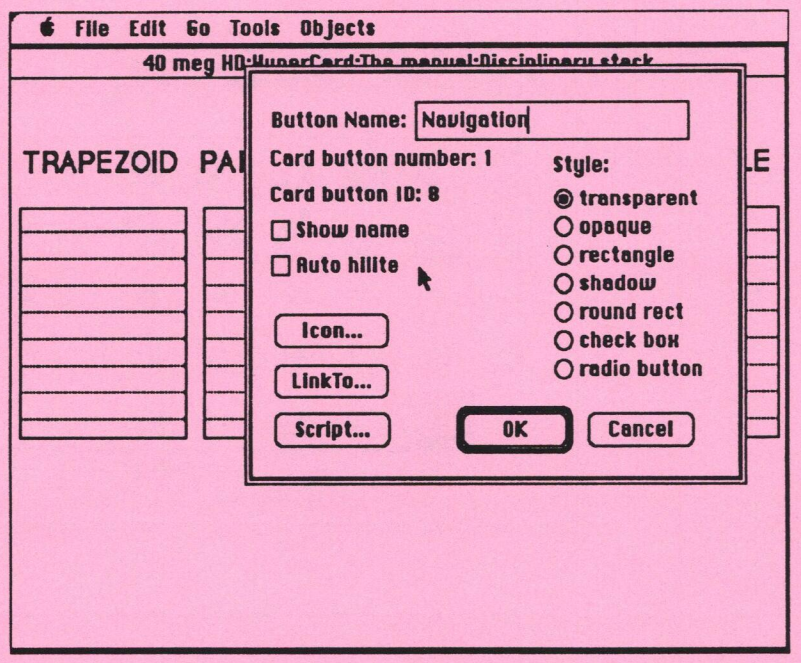


fig. 4 The Button Info dialog box allows for the style of a button to be selected or changed. Every button can be named and is numbered by the Mac so that it can be identified.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designedobject of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Icons Add Pictures to Buttons

Now click on the **Icon** button (on the left above the LinkTo button) and then look through the icon choices for the button you want. Choose the right direction arrow which is on the bottom row of the first page, the second one from the left. **(fig. 5)**

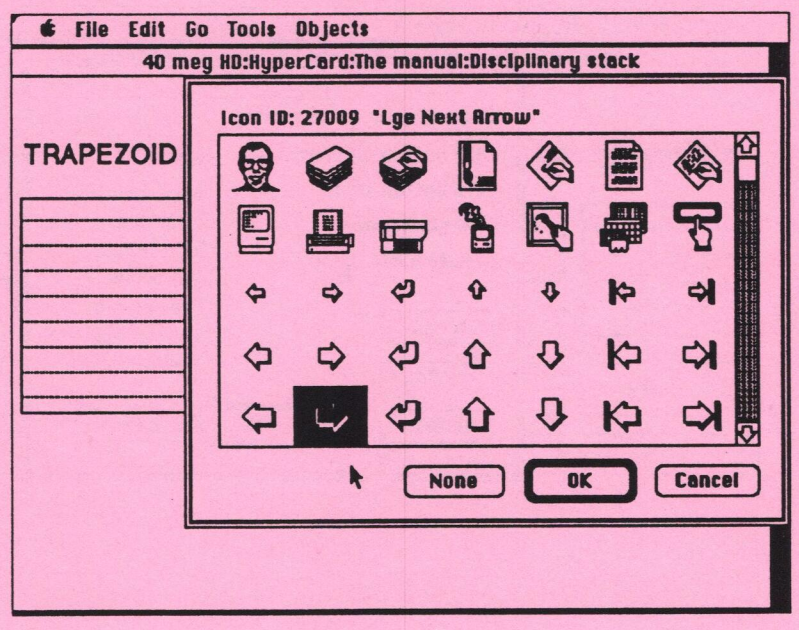


fig. 5 Icons or pictures may be chosen for a button. Simply select the desired icon by clicking on it. Then click on OK to confirm your choice. Icons may be removed later by clicking on None.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Choosing an Icon

Any of the other icons can be chosen to be the picture for the button, however, the idea of consistency is important in HyperCard design. HyperCard is very flexible which can be a cause of confusion. It is important to avoid overload with respect to the number of options available. You should feel free to experiment without feeling swamped by the choices.

Go back to the card level now and take a look at the button to make sure that it is in the right place and is the right size. Adjustment to size and placement can always be made. **(fig. 6)**

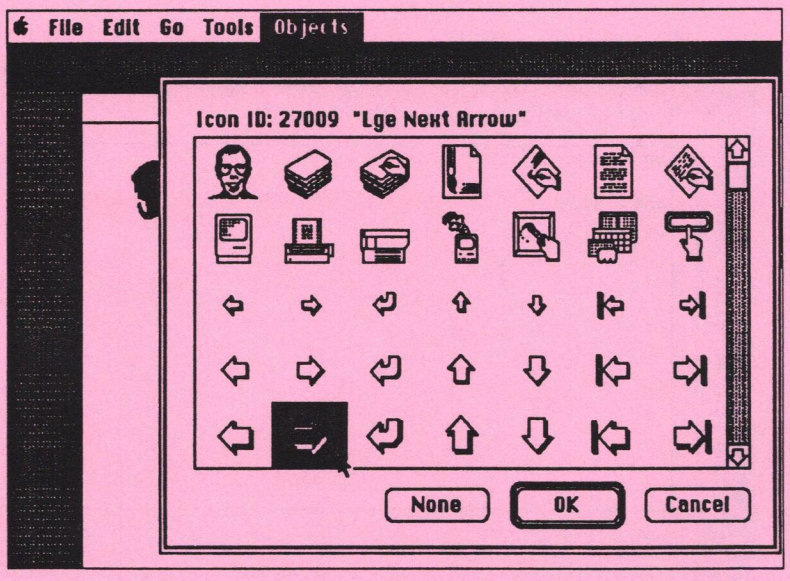


fig. 6 The Arrow button looks familiar and normal on the card level. The browse tool is used to put a button through its paces. The new button will not "do" anything until the instructions, or script, are added.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designedobject of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Adding the Script to a Button

Choose the browse tool, noting that you leave the background level and enter the card level when the browse tool is chosen. Now when you click on the arrow button, you notice that nothing happens. Nothing happens because the button has no script.

To write the script for the button, go to the **Button Info** dialog box on the **Objects** menu. A shortcut here is to click twice to get the Button Info dialog box. Now click on the **script** choice in the lower left corner of the dialog box. The screen will now show the script for the background button we have been working on. **(fig. 7)**

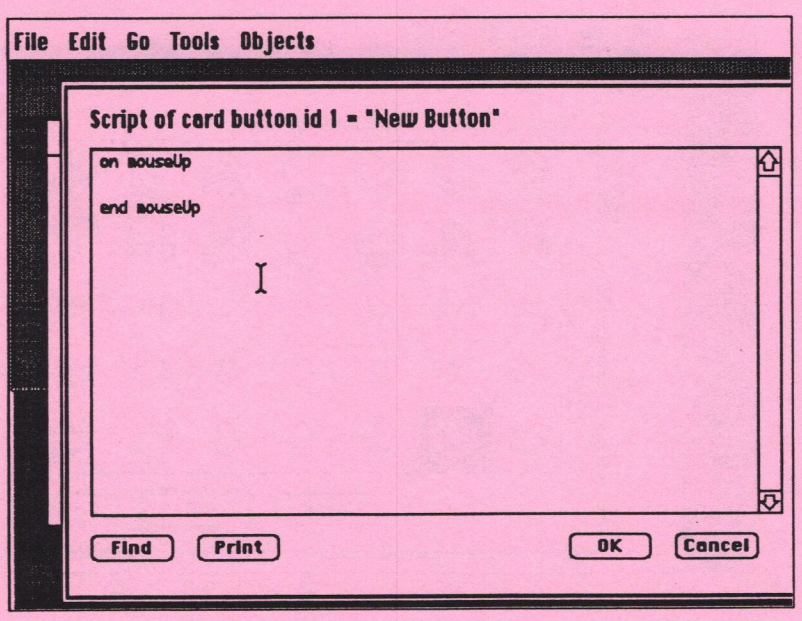


fig. 7 The Script for the arrow button is seen by choosing script on the Button Info dialog box. The script is partially written since HyperCard expects that a button will be clicked and therefore be activated by a MouseUp event.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Understanding 'on mouseUp'

The expression "on mouseUp" refers to the way that the computer looks at the user's operation of the mouse. When you click on something you push down. When you let up, the click is completed. The computer reads this completed click and performs whatever actions you have suggested between the terms "on mouseUp" and "end mouseUp."

At this point there are no commands between the "on mouseUp" and "end mouseUp" since none has been entered, but the cursor is blinking in the right spot, waiting for your entry.

Now we enter the script for this button, which is very simple, "go to next card." **(fig. 8)**

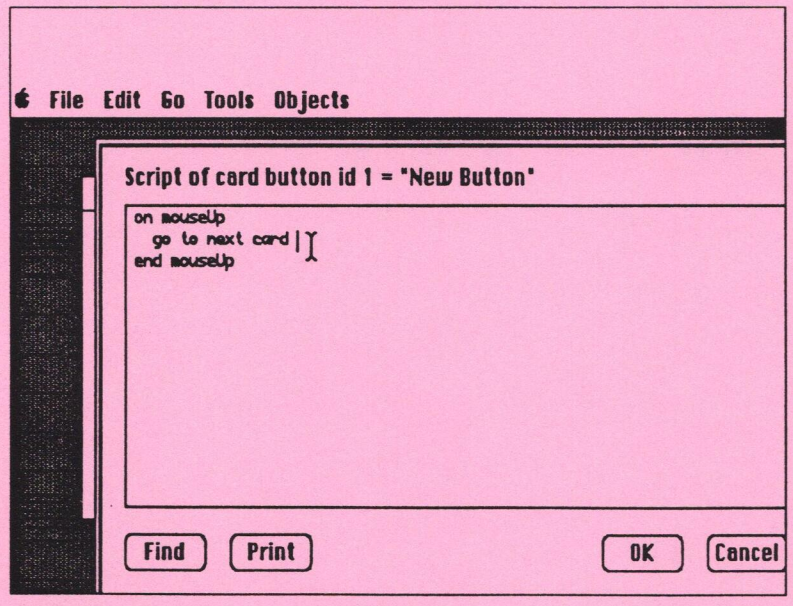


fig. 8 Enter the script for the button between the on mouseUp and end mouseUp. The script will be performed by the Mac when the mouse button is clicked.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designedobject of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Important note:

We consider it very important that this actual writing of the script be the first experience with scripting a button, as opposed to simply using the **Link To** choice on the button dialog box. It is no tragedy if you have already used HyperCard and are accustomed to using the **Link To** capability to make these simple buttons, but we are commited here to dispelling mysteries and eliminating the "black box" mentality.

Later, when scripting a button is well understood and hard links are being made between cards and stacks, using the **Link To** method will be the obvious way to go. Furthermore, our experience is that scripting makes sense, and it gives you a sense of control. We want you to feel empowered, so that you will feel good about empowering your students.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Making a New Stack
Adding Pictures from Other Stacks

Graphics are an important part of the impact of HyperCard. However most people are not accomplished graphic artists. HyperCard includes a stack of clip art from which we can copy pictures and then paste them into another stack.

We will now make a **New Stack** named **Copy Stack** without copying the current background. The first card is the title card. **(fig. 9)**

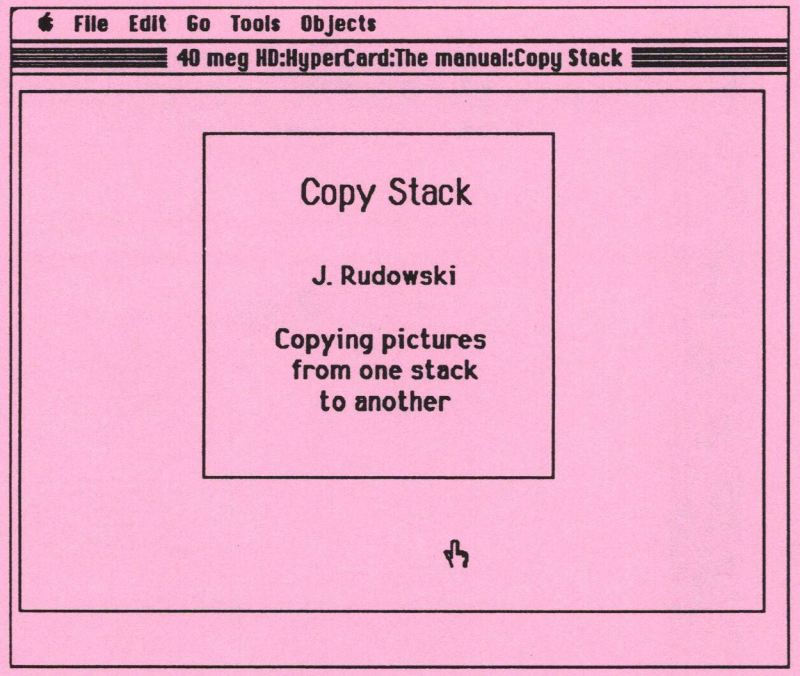


fig. 9 Make a New Stack using the New Stack choice on the File menu. Name the stack Copy Stack and do not copy the current background.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Find the Art Ideas Stack on the Disk

The pictures will come from the **Art Ideas** stack, where they should be **grabbed with the lasso tool, copied** and then **pasted onto this stack**. This will take a bit of practice if you are not very familiar with the Macintosh. A copy of Art Ideas has been provided on your disk to avoid some confusion. **(fig. 10)**

Learning how to lasso, copy and paste, and move things around to the spots you want them is an important skill to acquire in using HyperCard, so don't worry if it takes a few minutes to work this out. **(fig. 11a-e)**

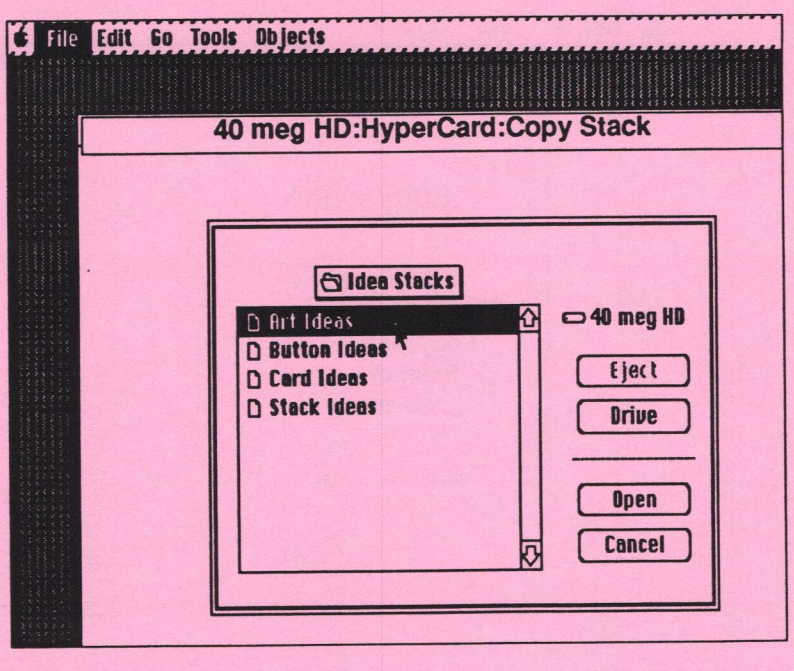


fig. 10 The steps to copy pictures from Art Ideas begin with choosing Open Stack on the File menu and finding the Art Ideas stack on the disk.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

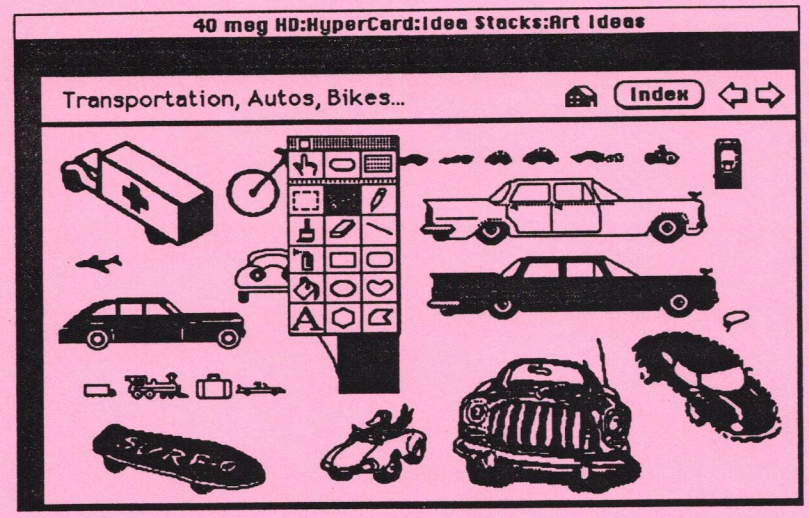


fig. 11a The Steps to copy pictures from Art Ideas: after opening Art Ideas, find the second page of the index and click on "Transportation, Autos, Bikes..." Using the lasso tool, select the car of your choice. Notice the dotted selection line around the car in the lower right of the screen.

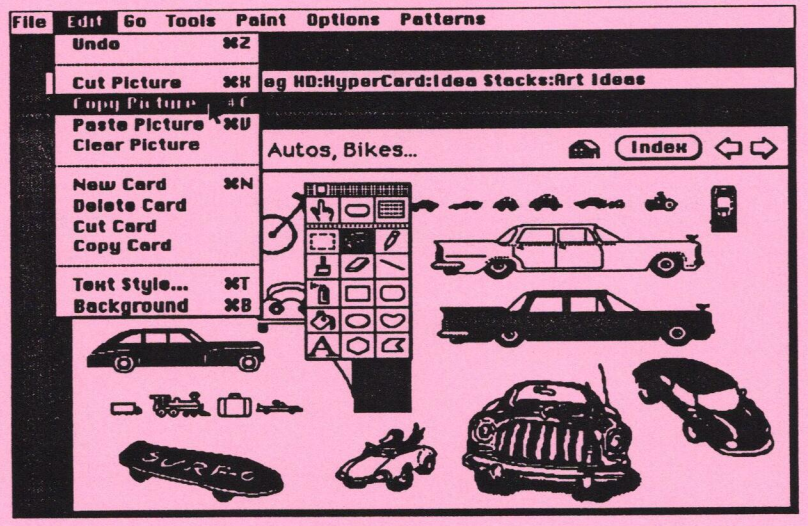


fig. 11b The Steps to copy pictures from Art Ideas: after selecting the picture you want, use the Edit menu to make a copy of the picture. A copy is put on the clipboard which is a temporary holding area in the memory of the Mac. It is better to Copy the picture rather than Cutting it so that the picture remains in Art Ideas.

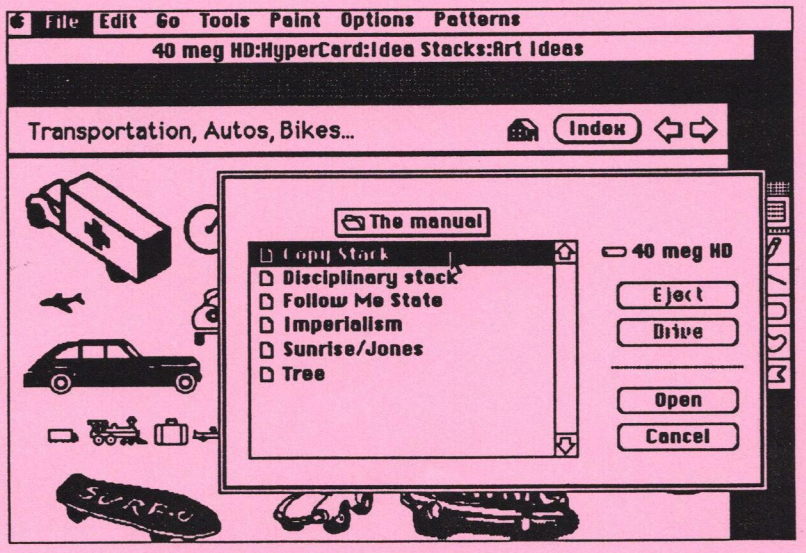


fig. 11c The Steps to copy pictures from Art Ideas: now re-open the Copy Stack by choosing Open on the File menu and finding the Copy Stack in the dialog box.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

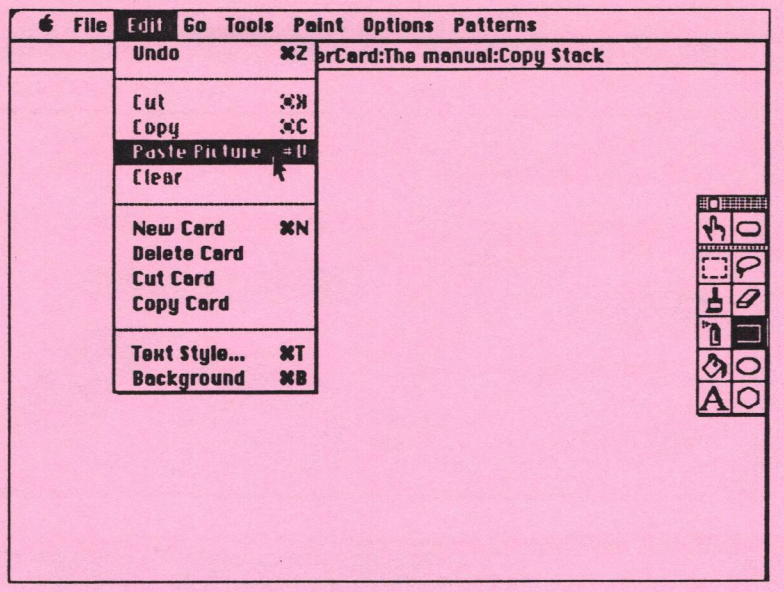


fig. 11d The Steps to copy pictures from Art Ideas: after returning to the Copy stack, select Paste Picture on the Edit menu to make another copy of the car from the clipboard into your stack.

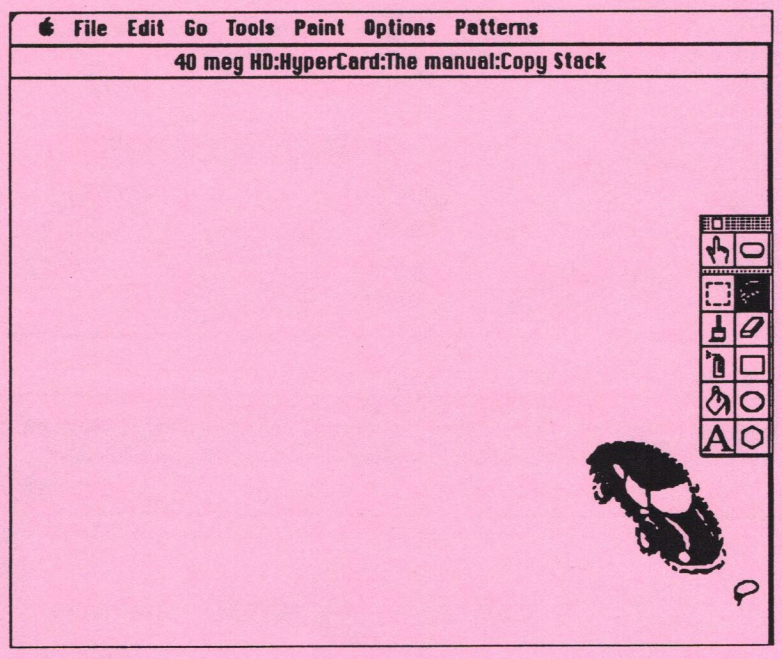


fig. 11e The picture of the car is now pasted from the clipboard on the stack. When the picture is pasted from the clipboard onto the card it arrives in the same place as it was on the previous card.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Transfer Art Work

Take some time to practice the technique of copying pictures from one stack to another. After a few minutes of practice time we will try out the **Recent** choice on the **Go** menu to navigate between stacks.

Select **Recent** on the **Go** menu and notice that the miniature cards which you see are all the cards which you have visited. They are "footprints" to use to trace the path of the cards that have been visited. Click on the small card to return to the full sized card of your choice. **(fig. 12)**

You may use this form of navigation especially when you are moving between stacks.

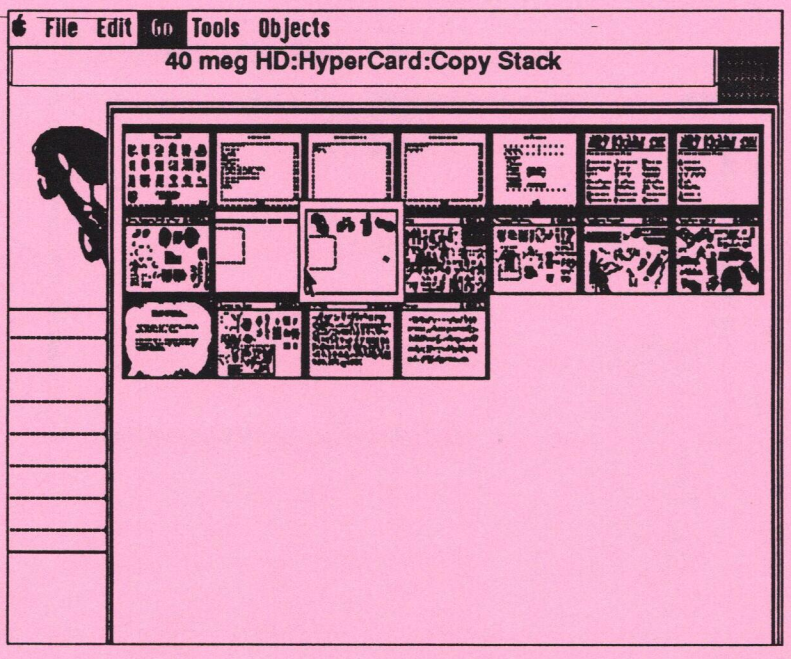


fig. 12 The Recent dialog box contains miniatures of the cards visited. Click on the small version of the card and the card appears even if it is on another stack.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Creating an Animated Stack

The main project of this session is to build an animation stack. In order to do this we will start with a follow me to **Import a MacPaint** picture and then use clip art from the Art Ideas stack.

Make a new stack named "Sunrise/(your LastName)" and toggle off the copy background option. **(fig. 13)**

You should now be on the first card of a new stack, with everything blank. This will be the title card. The process of creating a new stack with a title card first should be habit by now.

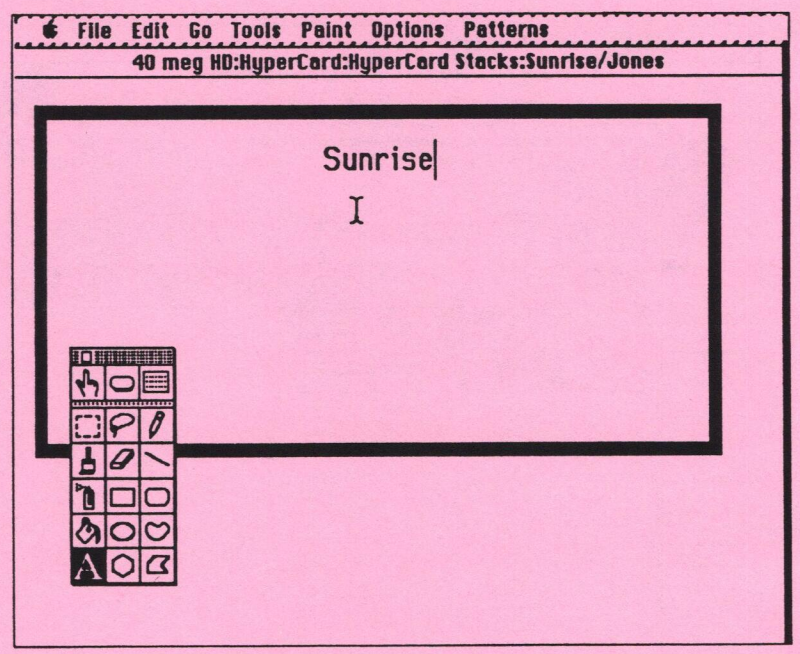


fig. 13 The Sunrise Stack will start with a rectangular frame around the card on the background of the first card.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Put a Frame on the First Card

Choose the rectangle tool (not dotted selection rectangle), get to the Background with a **command-B** and make a frame around the edges of this first card of the stack.

Designing the First Card

Now, back on the card level (toggle back to card level with a **Command-B**) using a large **text field**, fill in the information that we would like to see standard on a title card: Name, Title of Stack, Address, Phone number, and Date. You may change the fonts and styles for the field by selecting the **Text tool** and then pulling down the **Edit** menu and selecting **Text styles. (fig. 14)**

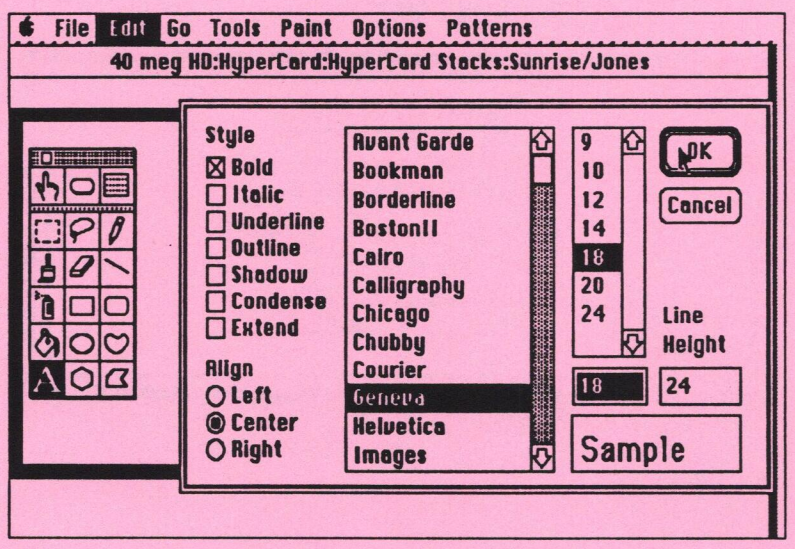


fig. 14 Text fonts, style and size can be changed by either selecting Text Style on the Edit menu or double clicking on the **"T" Text tool.**

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

The Title Card

Now you should have a new stack created and named with the word **Sunrise** hooked to your last name. The title card has a frame in the background and some standard information filled in on one large card level field. The title card for this stack is being designed completely in the background. This is important for what we are going to do next. **(fig. 15)**

HyperCard has the capacity to have several different backgrounds on the same stack, which can be confusing to people getting started with the program.

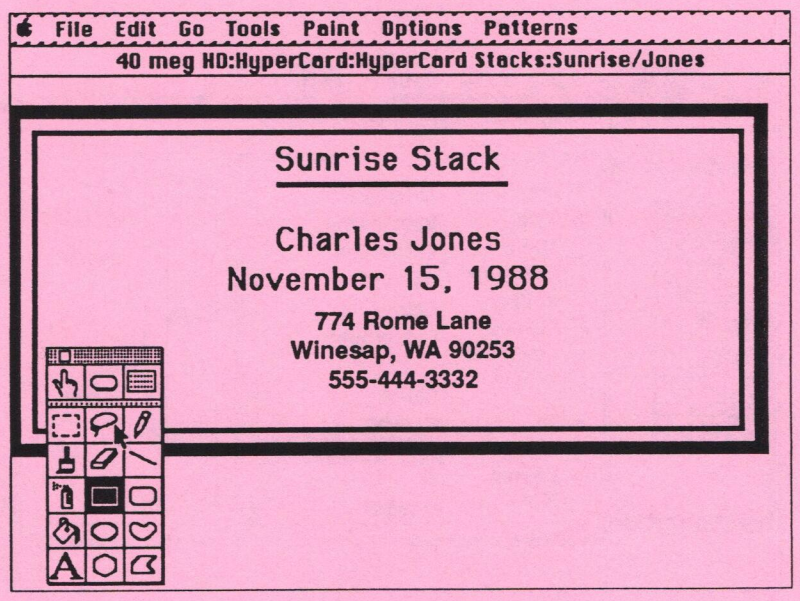


fig. 15 A Title card should be a standard design including the stack name, author's name, address, phone number, and the current date.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Break

Review

A great deal of material was covered in the first session. Take some time for review, think about the importance of being able to script a navigation button and grab pictures from another stack. You also learned a new way to navigate using the **Recent** option on the **Go** menu.

This is a good time to comment on ways to use HyperCard in the classroom. Remember the goal is to give you an adequate understanding of the power of HyperCard so that you will empower your students to use it to make the connections of learning. No longer should students be passive recipients of information when they can be directed to create knowledge with the teacher as a guide.

Objects	Button	Field	Script
Five objects are designed and scripted in HyperCard.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Using the Background in A New Way

In the discipline stack you were introduced to the background and card level format. We used only one background and opaqued the parts of the background that we didn't want to see on a particular card. That was fine for a starter, but HyperCard is much richer in its possibilites and often we will want the backgrounds for various parts of the stack to be so different that opaqueing will not be a good solution.

The concept of having one background for all the cards, then, is neat and easy to understand, but lacks dimension. With the possibility of many different backgrounds in the same stack, you can tie much larger concepts together, while still always having the one background per stack possibility available if that is sufficient. **(fig. 16)**

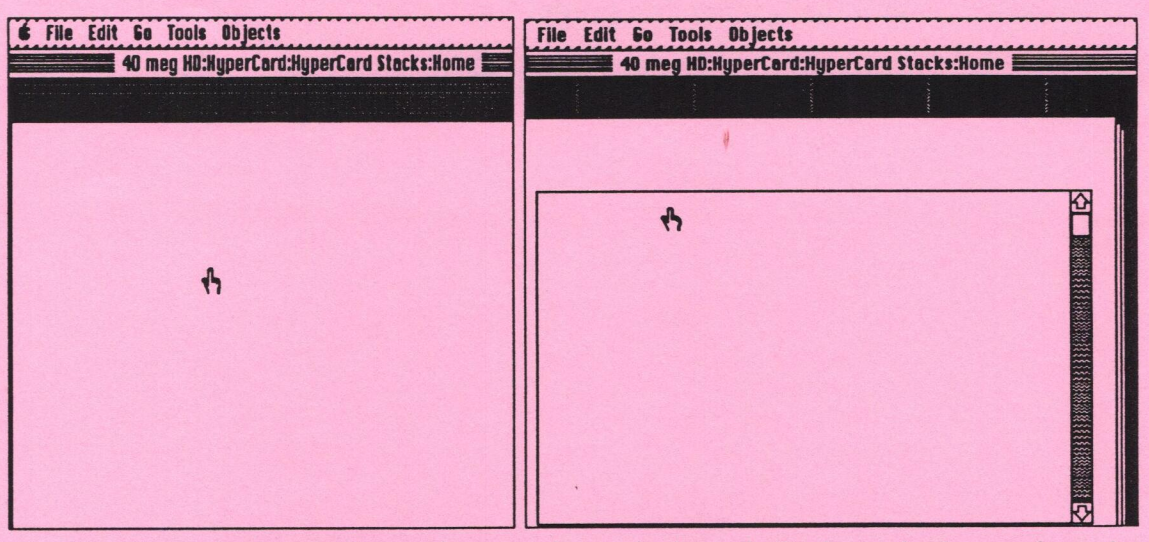


fig. 16 Different backgrounds may be created on a single stack. Each new background may be used several times and not necessarily in order. The Home stack contains 2 backgrounds as shown above. The stack info indicates the number of backgrounds.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Stacks may have Multiple Backgrounds

With multiple backgrounds, for example, a history teacher might have the scanned images of several different states used as backgrounds with each state forming the background for a series of cards about that particular state. One card might show the mountains and rivers, another the major cities, and a third might be shaded to show an economic or religious diversity. This group of cards might be followed by another set of cards dealing with another state.

You should be on your title card, and from there should choose the **New Background** option from the **Objects** menu. **(fig. 17)**

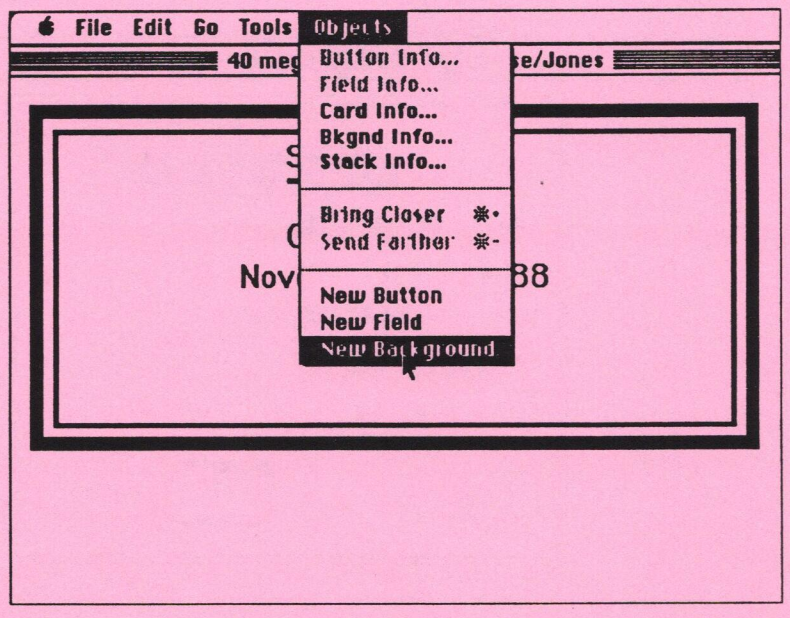


fig. 17 A New Background may be added to a stack by choosing New Background on the Objects menu. New Background adds a Card with a blank background.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Use Stack Info to Verify Stack Contents

Choosing a new background automatically creates a new card and moves to the new card. Now we will check the **Stack Info** choice on the **Objects** menu to confirm that this new stack now has two cards and two backgrounds, the title card and this new, blank card. **(fig. 18)**

Now you should navigate back and forth (using **Command-2 or 3**) from the title card to the new, blank card noticing that you have both cards and that the background (the frame) you made on the first card has not appeared on the second card.

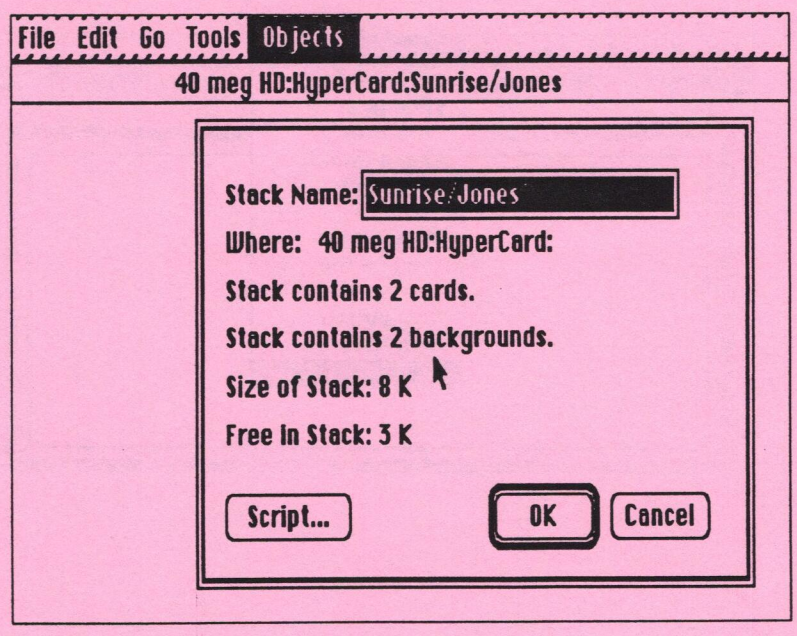


fig. 18 Stack Info on the objects menu now shows that the Sunrise Stack has 2 backgrounds and 2 cards.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Designing the New Background

Now go to the background of the second card. Choose the paint brush tool because the next operation we do will involve the paint tools.

We want to make a new background for this second card now and we want it to be a MacPaint picture called "Sunrise master" that is on the disk that came with this manual. We will "bring this drawing into" the background of this card by using the **Import Paint** option under the **File** menu. Notice that choosing the paint brush, or some other drawing tool, adds new options to the **File** menu, including **Import Paint. (fig. 19)**

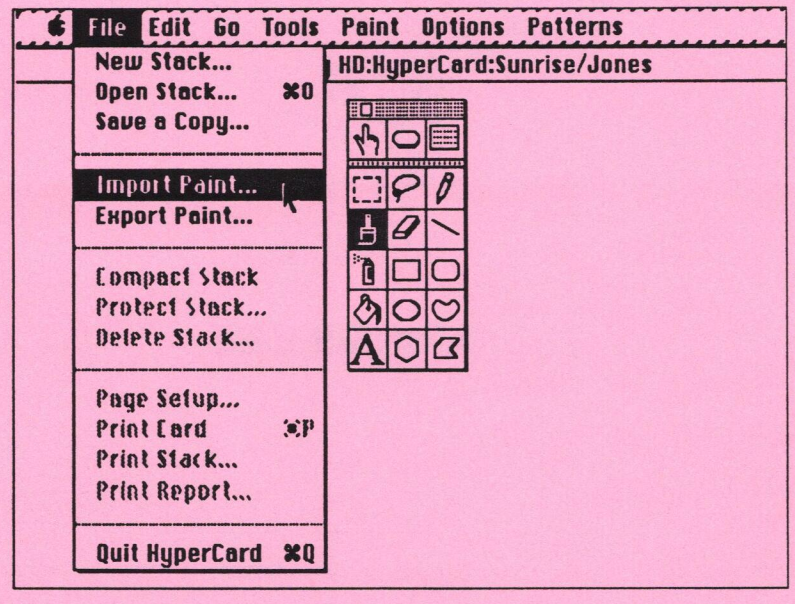


fig. 19 Import and Export Paint are only available on the File menu when a paint tool has been selected. Other menus change while using HyperCard depending on the tool currently chosen.

Objects	Button	Field	Clipboard
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	The clipboard is a temporary place in memory which holds one selection at a time. When an object or graphic is copied or cut it resides on the clipboard.

Import the Picture and Look at Both Cards

Choose the **Import Paint** option from the **File** menu and then look for the file called "Sunrise Master." **(fig. 20)**

When you have located **Sunrise Master**, double-clicking will bring it immediately into the background of the second card, if that is where you were when you selected the **Import Paint** option.

Take some time to practice going back and forth from card one to card two to check the backgrounds. Be careful that the drawing is in the background, not the card level.

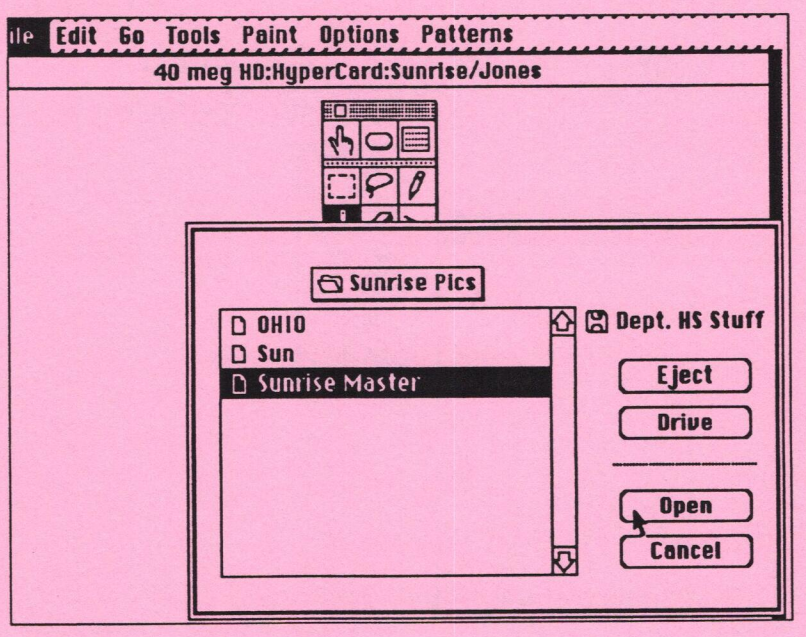


fig. 20 The dialog box for Import Paint allows the user to find the paint file desired. You may have to check the disk in an alternate drive or look through the folders on a disk to find the correct file.

Menus	Import	Background	Event
The HyperCard menus change with the tool that is chosen.	A command to bring graphics into HyperCard from a Paint file.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Explanation of the Sunrise Project

We are now going to create a simple **animation** exercise in the sunrise stack beginning with the work that is now completed.

The process will involve three steps:

1 - making a set of cards with two backgrounds, and importing a sunrise scene from a MacPaint document;

2 - dressing up these cards with navigational buttons;

3 - capturing a picture of the sun from the Art Ideas stack and installing it on each of the last six cards.

Objects	Buttons	Field	Clipboard
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	The clipboard is a temporary place in memory which holds one selection at a time. When an object or graphic is copied or cut it resides on the clipboard.

Step One

The first step is to make several new cards that have the same background as the one on your second card, namely the sunrise background that you have imported to that card. To do this we choose **New Card** from the **Edit** menu. This produces a new card with the same background as the card you are on when you make the menu choice. So card three, for example, will join the family with the same background as card two provided you are on card two when you choose **New Card**.

New Background ------> New Card **plus** New Background

New Card ----------> New Card with the **same** Background

Menus
The HyperCard menus change with the tool that is chosen.

Drag
Objects or graphics may be moved by pointing and dragging.

Background
The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.

Event
Clicking the mouse is an event which a script responds to.

Two Different Backgrounds on the Stack

Navigate through your stack now that you have three cards, noting the two different backgrounds.

Now repeat the **New Card** command four more times, so that you have a total of seven cards on your stack, the last six of which have the same background, the sunrise scene.

Now you should do some more navigating around the stack, and check **Stack Info** on the **Objects** menu to verify that you have seven cards.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Step Two

Now put navigational buttons on all seven of the cards, with no forward button on the last card and no backwards button on the first card. Notice again that the menus change depending on the tool and function being used. In this case the Copy, Cut and Paste functions indicate the possible choices. **(fig. 21)**

You may have more than seven cards in which case you can delete cards that are extra (Use **Delete Card** from the **Edit** menu).

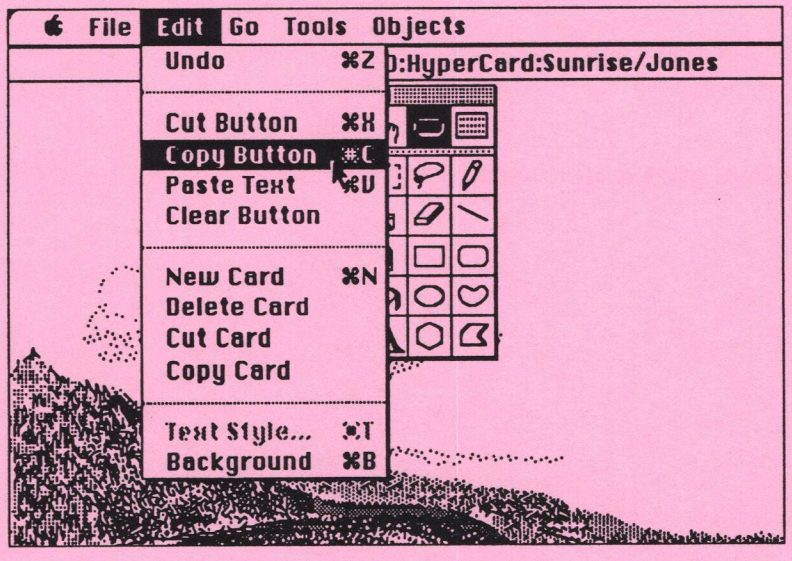


fig. 21 The Copy, Cut and Paste choices on the Edit menu change depending on the tool you are working with. For example, when you are using the Button tool the Edit menu choices change to Cut Button, Copy Button and Clear Button.

Menus	Copy	Background	Event
The HyperCard menus change with the tool that is chosen.	A command used to make a copy of whatever is currently selected.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Step Three

The goal now is to bring in a picture of the sun and place it in different positions on each of the cards two through six, moving it up a little on each card. Navigating through the cards one after the other will give the impression of the sun rising, more or less.

Import Graphics from a Variety of Sources

The picture of the sun we will use is in the **Art Ideas** stack. You should be able to appreciate the goodies available in the product. These ideas, images and icons can make your teaching applications much more simple. Furthermore it is a good idea to see different ways to "import" things into your stacks.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Finding a Picture from Another Stack

To get to the image of the sun that we want, go to the **Open Stack** option on the **File** menu. Find the copy of the Art Ideas stack and open it. You may have to do some "looking around" on the list of stacks available to find the Art Ideas stack. **(fig. 22)**

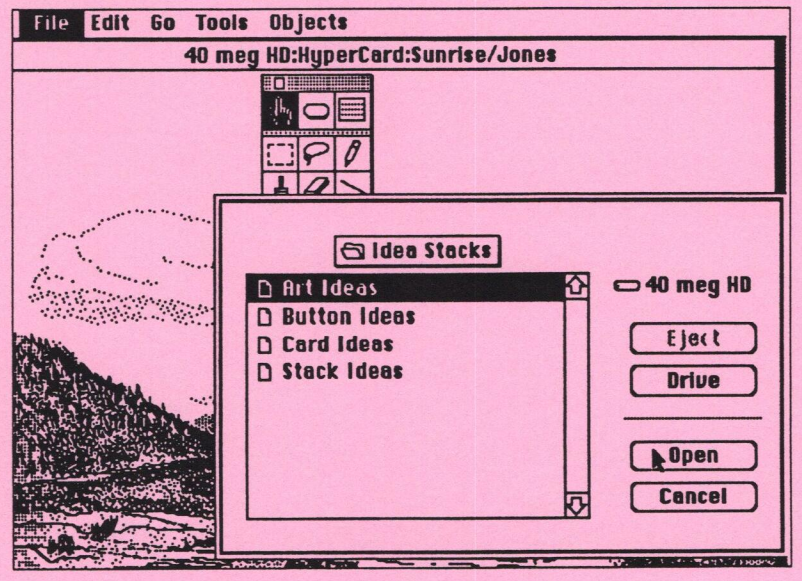


fig. 22 The Open Stack dialog box displays the stacks available on a disk. Using the Open button to open folders and the Drive button to move between drives, it is possible to locate the desired file. Having found the Art Ideas stack use the Open button to open the stack. All the tools and functions of HyperCard apply to this stack.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Finding Page Two of the Art Ideas Index

Assuming things have not been moved around a lot on the disks, the stack we want is called "**Art Ideas**" and is located with several other **stacks** in the **folder** named "**Ideas Stacks.**"

Click the next card button in the upper right in order to get to the **second page** of this index. Go back and forth a few times between these two cards with the navigational buttons. Now settle in on page two of this stack. **(fig. 23)**

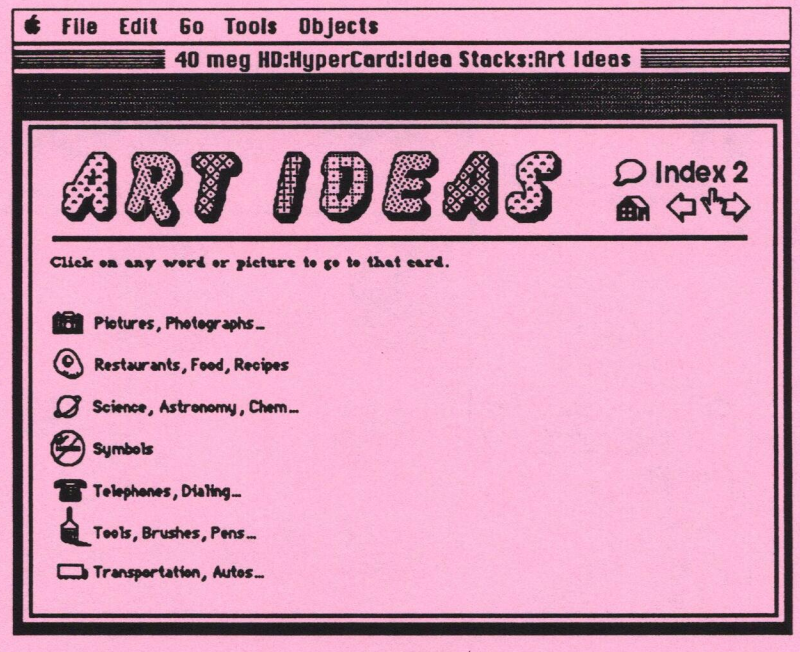


fig. 23 Page 2 of the Art Ideas index has the remainder of the clip art arranged alphabetically. Use the arrows in the upper right corner to navigate between the pages of the index. Find "Science, Astronomy, Chem".

Objects	Button	Repeat	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	The command which provides for repetition of a set of commands.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Choose "Science, Astronomy, Chem..."

Choose the listing entitled **"Science, Astronomy, Chem..."** and when this card is on the screen, notice the three versions of the sun at the top of the card. Select one - one of the larger versions looks best for this exercise - by using the **Lasso tool** from the **Tools** Palette.

When the sun has been selected it gives the characteristic blinking that indicates it has been lassoed. Choose **Copy Picture** from the **Edit** menu to store this picture on the clipboard for later use. **(fig. 24)**

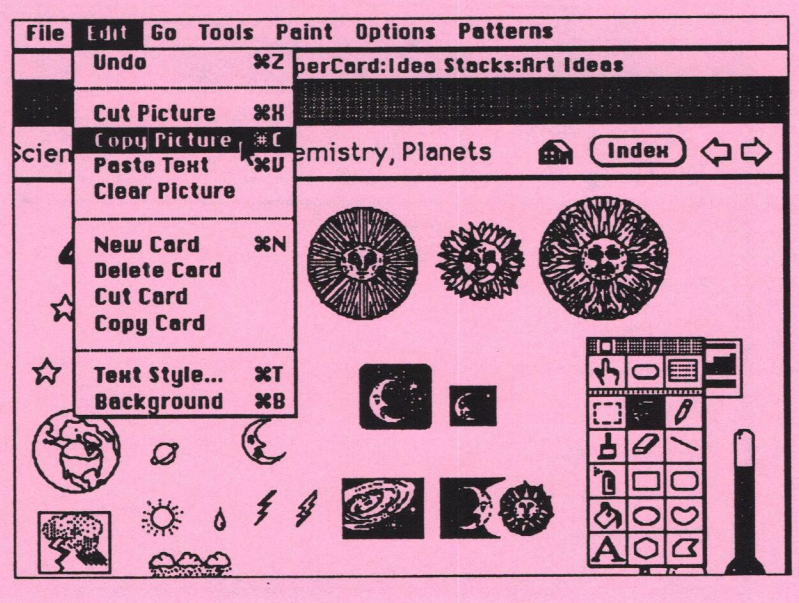


fig. 24 The Copy Picture choice on the Edit menu puts a copy of the selected picture on the clipboard. The clipboard is a temporary memory location which can hold only one item at a time.

Menus	Wait	Clipboard	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command which delays the action of a script.	The clipboard is a temporary place in memory which holds one selection at a time. When an object or graphic is copied or cut it is put on the clipboard.	The end of an event is noted with an end statement.

Copy the Picture and Go to a Recent Card

Now we return to our sunrise stack, this time using the **Recent** option on the **Go** menu, picking the card that has our sunrise picture already installed. You should navigate through the stack until you are on the card level of card two, the first card on which the sunrise picture was installed.
(fig. 25)

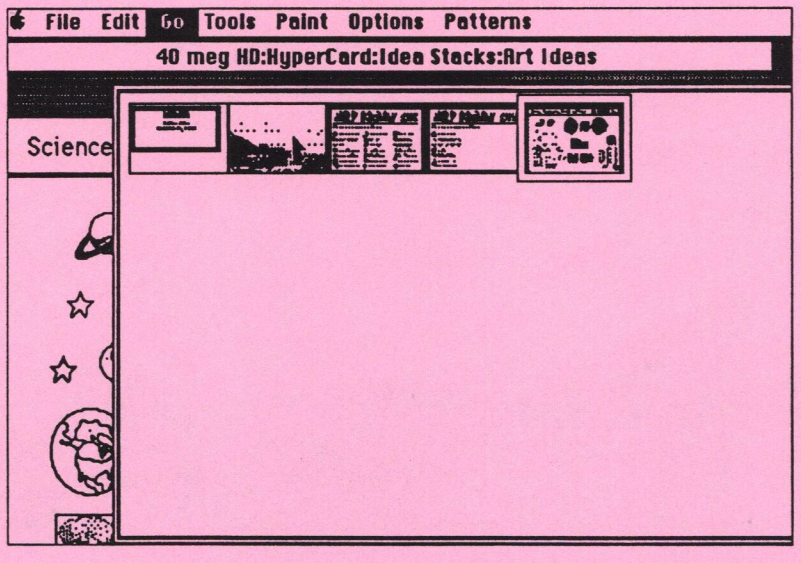


fig. 25 The Recent option on the Go menu shows us miniatures of the last cards to which we have navigated. Click on the card you want to "go" to and you're there even if it is on another stack.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Paste the Picture into the Sunrise Stack

Choosing **Paste Picture** from the **Edit** menu will paste a copy of the picture of the sun on the card level of card two. The sun will still be "blinking" around the edges, indicating that we can move it around as desired.

Erase Part of the Picture to Reveal the Horizon

Install the sun overlapping the horizon and then choose the **eraser tool** and erase the bottom part of the image to show the background picture and give the impression of the sun being behind the horizon. **(fig. 26)**

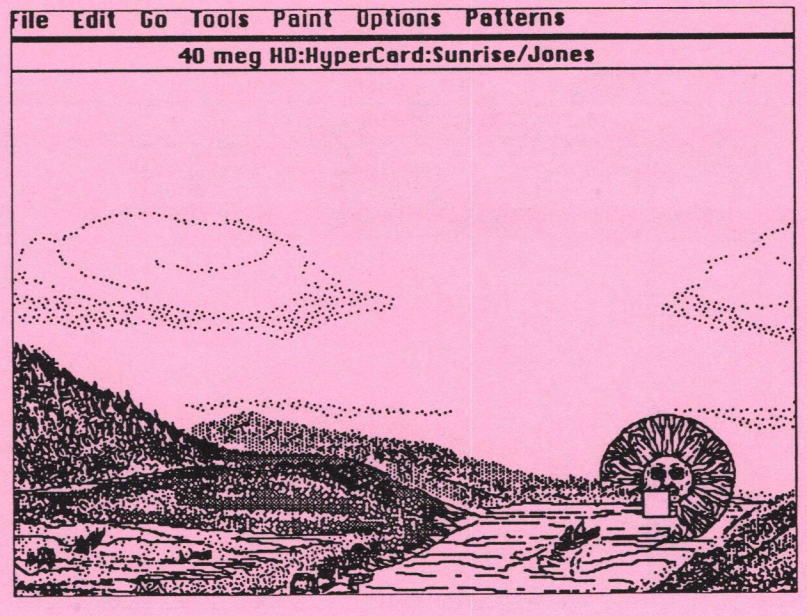


fig. 26 The Eraser tool is used to erase the part of the sun which should be beneath the horizon. Erasing the sun on the card level reveals the scenery on the background.

Menus	Drag	Background	Event
The HyperCard menus change with the tool that is chosen.	Objects or graphics may be moved by pointing and dragging.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds

Complete Successive Cards to Make the Sun Rise

Now go to card three and paste the same picture in and move it to a spot slightly higher than the original, perhaps still erasing some of the bottom of the sun to make room for the horizon. On each successive card, paste in the sun and move it up slightly till it reaches a zenith on card seven.

Time to Complete the Cards

You will need some time to work on this. Having the sun move up into the sky as you move from card to card, is exhilarating. It is a degree of control over the computer that has been simply unavailable to people not expert with a computer language.

If you select the sun and copy it from it's place on one card and then paste it onto the next card, it will arrive in the same location as on the previous card. Now you can judge accurately how much to move it in the sky.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Adding Action to the Stack

A final exercise for this module would be to have the cards "play themselves" by having a button on the title card that will have HyperCard go through the remaining cards. We can do this with a button. Create a **New Button** by choosing the option on the **Objects** menu. Change the name and style of the button as desired. **(fig. 27)**

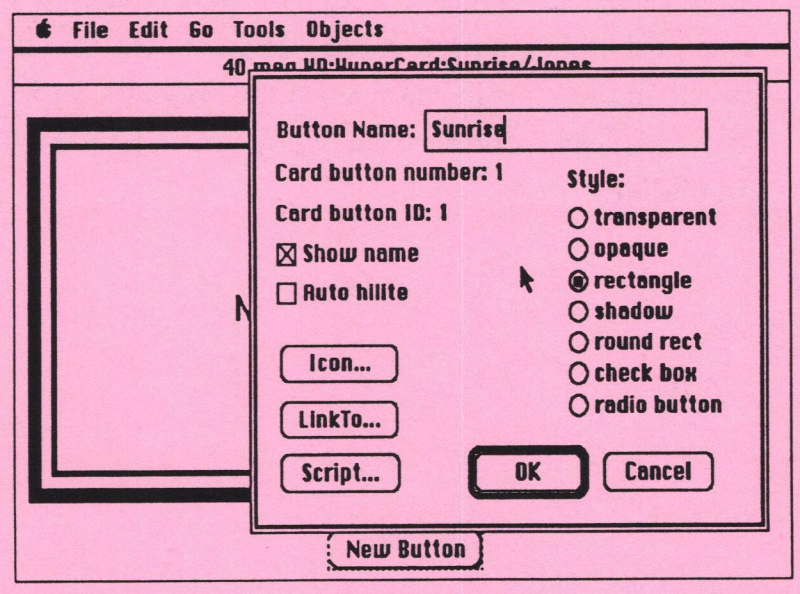


fig. 27 The Button Info dialog box has choices for changing the style, name and icon of a button. Also notice the numbers corresponding to the button. Buttons often need to be specifically identified. The script option shows the box in which script is written.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to display cards, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Write the Script to See the Sunrise

Choose **Script** from the **Button Info** dialog box and complete the script to show the 6 cards in the stack. **(fig. 28)**

```
on mouseUp
    show 6 cards
end mouseUp
```

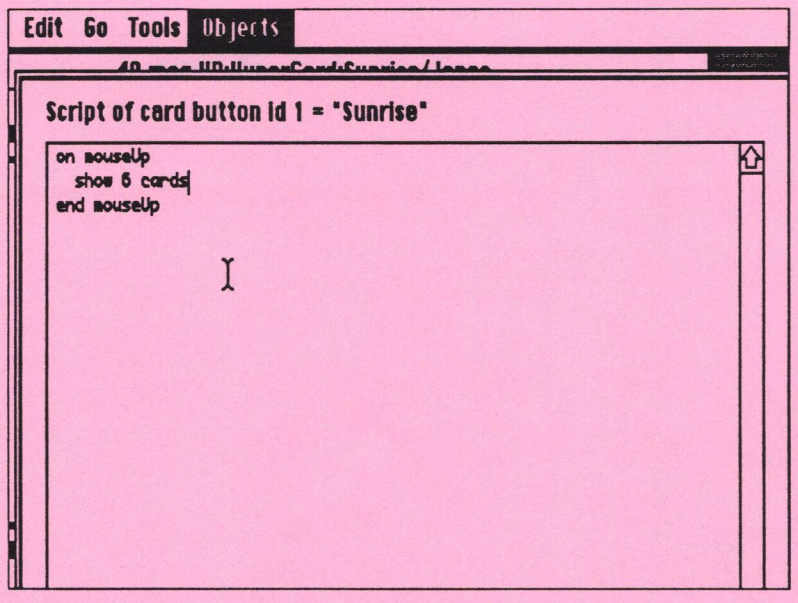


fig. 28 The script for the sunrise button is written to show all the cards in the stack. The Mac manages the time between cards. Visual effects may be added to the script to change the appearance of the screen between cards.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Revised Script for the Sunrise Button

This runs too quickly to appreciate our sunrise, of course, but it is a simple command that you may find to be of value. A better script for the button, accomplishing the same thing, at a more reasonable pace is:

```
on mouseUp
    repeat 5 times
        go to next card
        wait 30 ticks
    end repeat
end mouseUp
```

(fig. 29)

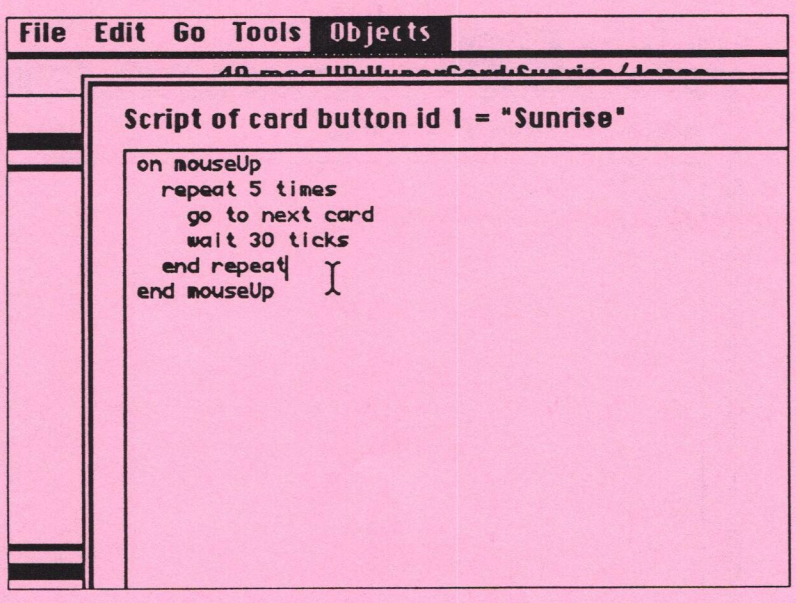


fig. 29 The Script for the button may be modified as shown to make the sun rise more slowly. HyperCard allows time to be counted in seconds and partial seconds called "ticks". There are sixty ticks to one second.

Menus	Ticks	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk measure of time ; there are 60 ticks in one second.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

Repetition is Possible with a Repeat Loop

This script introduces the idea of a loop, but kind of in disguise. You may not even realize that you are doing it because the HyperTalk has such close English equivalent.

Ticks are parts of seconds which can also be used with the **Wait** command, with 60 ticks equalling one second. Experiments can be done with the timing to get it just the way each person wants. The "end repeat" statement is not naked programming, but isn't English either. It can be explained quite simply as the marker for what it is that you want repeated five times.

Objects	Button	Field	Script
Five objects are designed and scripted in Hyper-Card.	The user designed object of navigation and connection.	Fields may contain numbers or text and may vary in style.	A script is the programming element of HyperCard which adds control of the objects. Scripts are written on the objects and respond to events such as mouseUp.

Conclusion and Review

We will conclude the module with review of the new skills introduced. The notion of animation is one that demonstrates the power of HyperCard to produce a fairly dramatic presentation in a short time. More importantly the concept of repetition is a staple of computer science. The ability to repeat a set of commands is essential to creation of simulations as well as animation. This concept will be built on in the final module.

Review also the idea of Background as a separate level from the card. Think about your impressions about the usefulness of having a level which is protected from erasure.

Menus	Show	Background	Event
The HyperCard menus change with the tool that is chosen.	A hypertalk command used to reveal graphics, text, or buttons.	The background is the bottom level of a card. There may be more than one background in a stack. The background is designed separately from the card level.	Clicking the mouse is an event which a script responds to.

HyperCard in Education

Module 4 Participant's Guide

Joseph F. Hofmeister • Joyce B. Rudowski
The HyperSchool
Cincinnati Country Day School
6905 Given Road, Cincinnati, Ohio 45243
513-561-7298 AppleLink K0197

Module 4

Introduction
Considering HyperCard in the Classroom

Beginning this last of four modules keep in mind the overall goals of this training, to put a great new mind tool into the classroom environment. Each of you trained to use HyperCard must be sufficiently comfortable with it to be able to devise general ways in which it can be used in the classroom. The essence of the whole business, however, is to empower the student to become more involved, more interested, more excited and more in control of the learning process. HyperCard can be used as a fine presentation tool, but if that's the only way it is used, its real impact in classrooms will be largely missed.

Objects	Message	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	A message is the cause of interaction in HyperCard.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Ownership of Learning

Students who "build" Africa on HyperCard, with buttons that link information-laden cards, which the students themselves have prepared, will know, remember and even discover things about Africa that they will not take with them from even a very good presentation by a teacher. Students take ownership of information and connections in a new way when they do it themselves. **(fig. 1)**

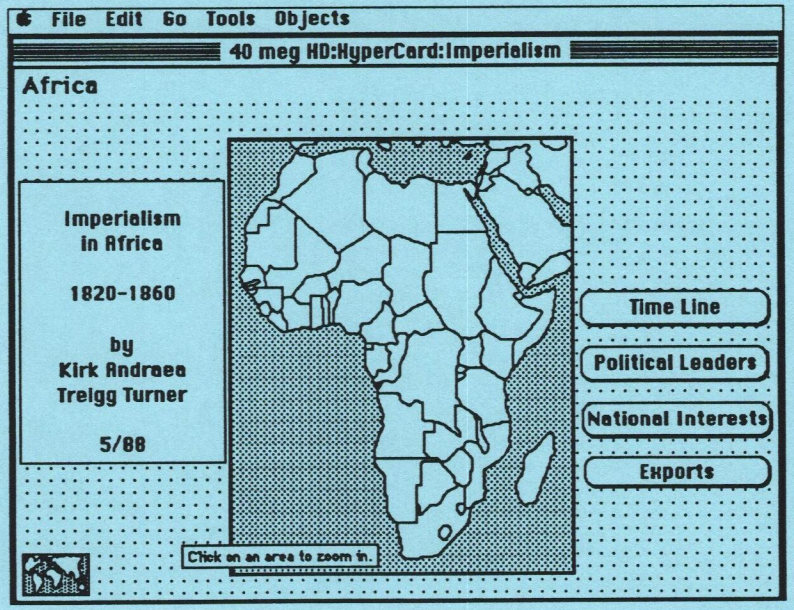


fig. 1 A student created project on Africa means more than the very finest lecture on the same subject. Buttons act as a dynamic table of contents connecting information cards, maps and analysis.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Looking for the Possibilities

It is more important that you take ownership of the program than that you know what's in every nook and cranny. It is far more important for you to understand the possibilities of HyperCard in your subject area than to know every bit of scripting.

No matter what you do in these four modules, sharp, slippery minds among their students will soon discover lots about HyperCard that the teacher didn't know when the program was introduced. Students who are already familiar with programming languages will react like kids turned loose in a candy shop. Keeping everyone in the classroom at the level of expertise of the teacher is an exercise in futility, accomplished only with marginal probability at best and with severely limited access to the computers a paradoxical requirement.

Objects	Message	Handler	Connections
Five objects are designed and scripted in HyperCard.	A message is the cause of interaction in HyperCard.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

The Kids are so Comfortable with the Computer

Other computer programs have this same kind of flavor. You bring them into your classroom and even though you have worked on them over the weekend or for a few weeks, you still find your students teaching you something about them. HyperCard, however, extrapolates this situation to some kind of a limit. The program is so rich, so diverse in application, that student surprises are inevitable. The idea is one that many teachers will feel a little queasy about. It isn't necessary or even desirable that they know every possible step in how one might want to get there. Setting directions and seeing where a project is going is the point in this module's first exercise after our regular beginning. **(fig. 2)**

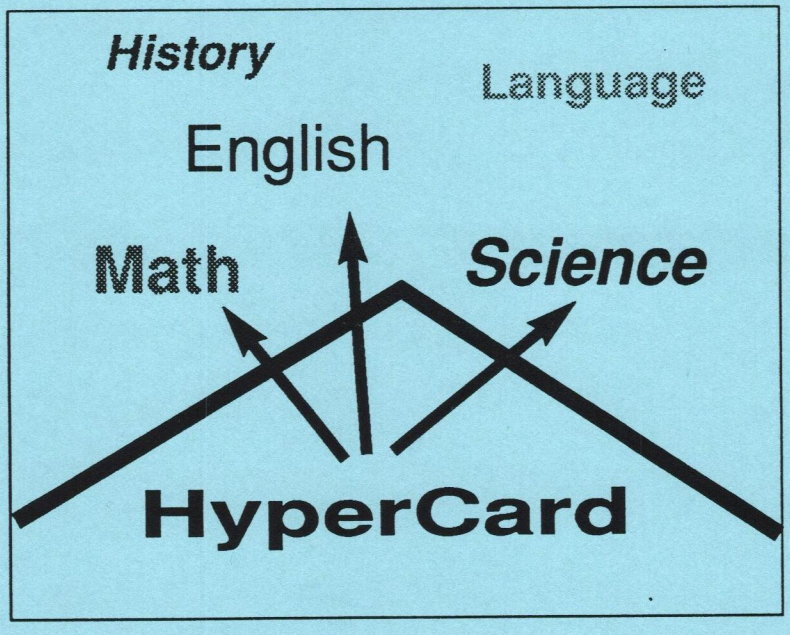


fig. 2 HyperCard takes to new horizons the fact that students will discover in the program ideas and applications that teachers will not think of.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Review and Discussion

We will begin this module with the standard question and discussion period. In the last session you developed a smooth animation sequence. This demanded a fair amount of familiarity with the Macintosh clipboard and the copy and paste functions as well as some experimentation with various time lags in a **show next card** scripting sequence. What are the questions that remain from that session? Some of you may be asked to demonstrate your stacks if time allows. **(fig. 3)**

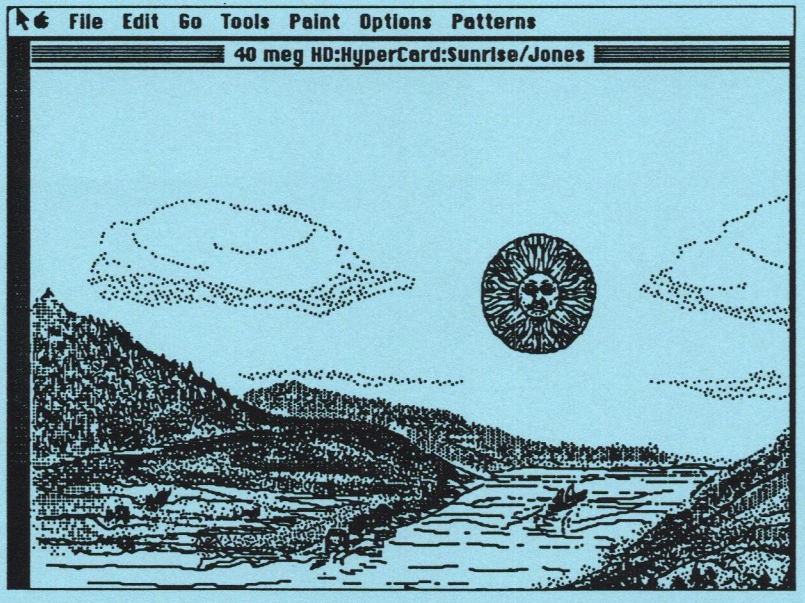


fig. 3 The animation stack provided a look at moving graphics into HyperCard stacks as well as scripting.

Objects	Import	Handler	Message
Five objects are designed and scripted in Hyper-Card.	Import Paint is a way to incorp-orate graphics into a stack.	Scripts are said to handle the events of the HyperCard objects.	A message is the cause of interaction in HyperCard. Messages come from the message box, buttons and the menu options.

Small Group Discussion

The next exercise calls for the class to be broken up into groups of four or five each. There will be one leader for each group. The groups should be separated in the room or put in different rooms if the space is not adequate to distance them from one other.

This is a structured exercise providing limited time to increase the notion of response rather than well thought out ideas.

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Event
Clicking the mouse is an event to which a script responds.

Forming Ideas on the Use of HyperCard

The object is to have each participant come up with some specific use of a stack in his or her discipline. In module two, we did an elementary stack in varying disciplines but that was simply to give you a feel for the nature of a stack, with cards that have backgrounds and card levels. The stack ideas suggested now should be much more curriculum centered. They should be stacks that your students might be asked to create as part of an assignment.

You will be asked to tell the rest of the group about a stack that could apply to some individual subject area. These should be practical, concrete ideas. They do not have to be earth-shaking creations, but rather minor, perhaps even unimportant applications of HyperCard.

Objects	Click	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The click of the mouse is an event which causes action.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Significant Ideas will Emerge

Some of the stack concepts that come out of these groups will definitely not be minor, but no one should be expected to come up with anything spectacular. Leaders in each group will start the ball rolling themselves by coming up with a stack that applies to some discipline.

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Event
Clicking the mouse is an event to which a script responds.

Time Limits for this Process

There is an absolute time limit on the description of each idea. This time limit to present the stack idea will be two minutes maximum.

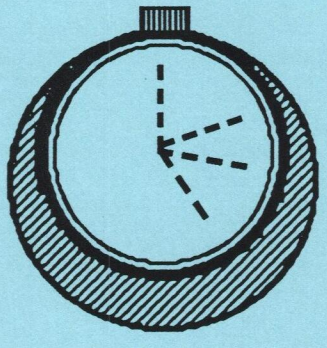
The idea doesn't have to be complete but can be just a basic framework, the mere shadow of a stack idea. Hold to the two minute time limit!!

Objects	Click	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The click of the mouse is an event which causes action.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Responding to Ideas
Participants React Positively to Stack Ideas

After the stack idea has been described briefly, the rest of the participants in the small group will each give their response to the stack. Each response must be short (one half to one minute maximum) and each response must absolutely be some positive extension of the concept.

This is not the time when the idea is evaluated or criticized but rather given that someone has given a two minute summary of a stack idea, each other person in that group then gives a 30 to 60 second response to the idea. **(fig. 4)**



Time Limits:

Presentation:
2 Minutes

Reponses:
30-60 seconds

Summary:
1 minute

fig. 4 The time limits for this exercise will be strictly enforced. The purpose of the exercise is to brainstorm ideas without concern for detail or logistics.

Hide	Show	Scripted Action	Event
A hypertalk command used to remove an object from view.	A hypertalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event which a script responds to.

Responses which Build on the Idea

The response must be couched in terms like: "Well, another thing you might think about in that stack is to do this..." or "That's a neat notion and I think it would be fun to include the idea of ..." No one is allowed to say something like "Actually, I don't think that would work as well as ..." Each person listening to the idea takes the idea and adds a piece to it.

Objects	Click	Handler	Connections
Five objects are designed and scripted in HyperCard.	The click of the mouse is an event which causes action.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Hold to the Time Limits

As intimidating as the affair may sound it is a powerful method. Not only do you present some simple idea of your own and then get some goodies added to it by each member of the group but you also hear a lot of other original ideas which you participate in by chipping in with creative ideas of your own.

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Event
Clicking the mouse is an event to which a script responds.

Gather the Whole Group For Review

The group will gather back together after this exercise. It may be the time for the break considering the time involved. Furthermore you may be anxious to talk to each other about the ideas suggested during the small group session.

Objects	**Click**	**Handler**	**Connections**
Five objects are designed and scripted in Hyper-Card.	The click of the mouse is an event which causes action.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Focus on Scripting

The next project will involve more scripting. This series of projects will be a reinforcement of many of the skills learned in the previous sessions and add some new skill concepts.

We will begin with two Follow-Me exercises. The first will involve creating pop up fields on a new stack. The second will involve script which will draw a line on the screen from one predetermined point to another.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

New Skills to Learn

There are three new operations that you will need help with before you can take off on the next project:

1. **Making a pop up field.**
   A pop up field is a field which is controlled within a script. The field will be made visible for a time and then disappear from view.

2. **Choosing a tool from the palette within a script.**
   It is possible to include in a script a command to select any tool on the palette. This ability may be used to create animation by writing a script that will select a picture on the screen and then drag it from one place to another.

3. **Dragging from one point to another on the screen. Just as it is possible to choose any tool from the palette, it is possible to move the tool from one point on a card to another. The result is that a line or square or oval will be drawn on the card.**

Objects	Pop up	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	Something which suddenly appears on the screen.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Follow Me to Pop Up Fields

We will demonstrate the new operations with a Follow Me exercise and then you will be turned loose with a set of steps to follow which will require that you refer back to the Follow Me or remember how to do it.

First you will create a **New Stack** naming the stack **Pop Up Stack**. You should not copy the current background. Complete the standard title card and then add a new card. **(fig. 5)**

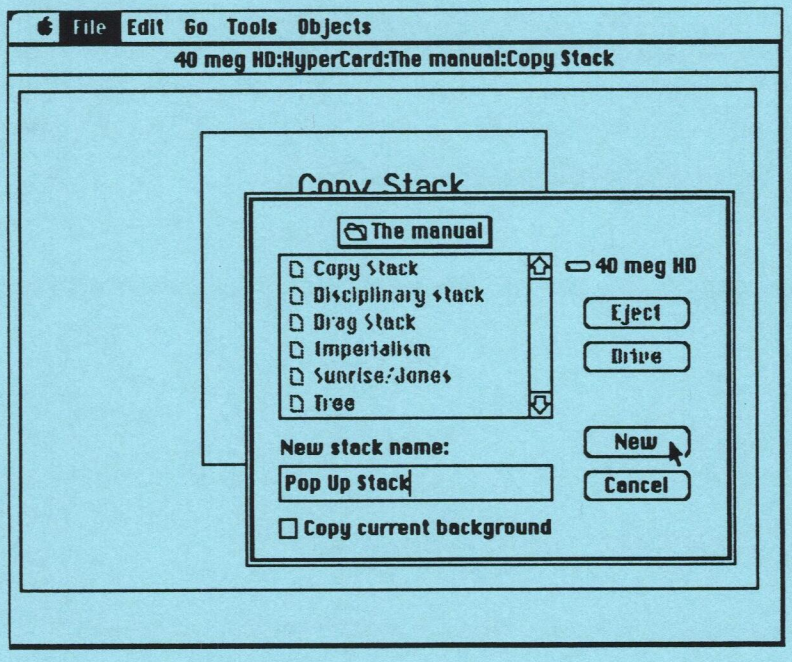


fig. 5 This follow-me exercise starts with making a New Stack called Pop Up Stack. The stack will have a title card and then a second card with a text field and a button.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Add a Field to the Stack

Now make a field on the second card showing the lines and making it a rectangle or shadow. Returning to the browse mode write a phrase, some names or comments into the field.

Together we will get Field Info and notice the **card field number** and **card field ID.** In the script we will write we will need to refer to this card field specifically. **(fig. 6)**

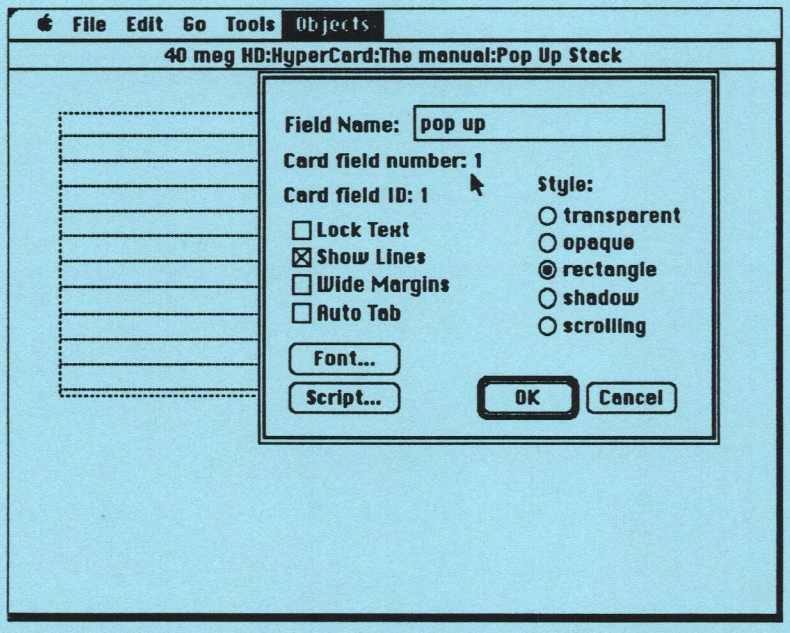


fig. 6 The new field on the Pop Up Stack has a unique number which identifies it from any other field on this card. We will use the number to identify the field in the button script.

Objects	Drag	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The command which causes a tool to be moved on a card.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Scripting a Button to Pop Up a Field

Now create a button and move it to the top center of the screen out of the way of the field. Get **Button Info** and change the name of the button to Pop Up. Go to the script for this button and write the following code:

```
on mouseUp
    show card field 1
    wait until the mouseClick
    hide card field 1
end mouseUp
```

(fig. 7)

Try out the button several times until you are comfortable with the script. You will have to click once to show the field and once more to hide the field.

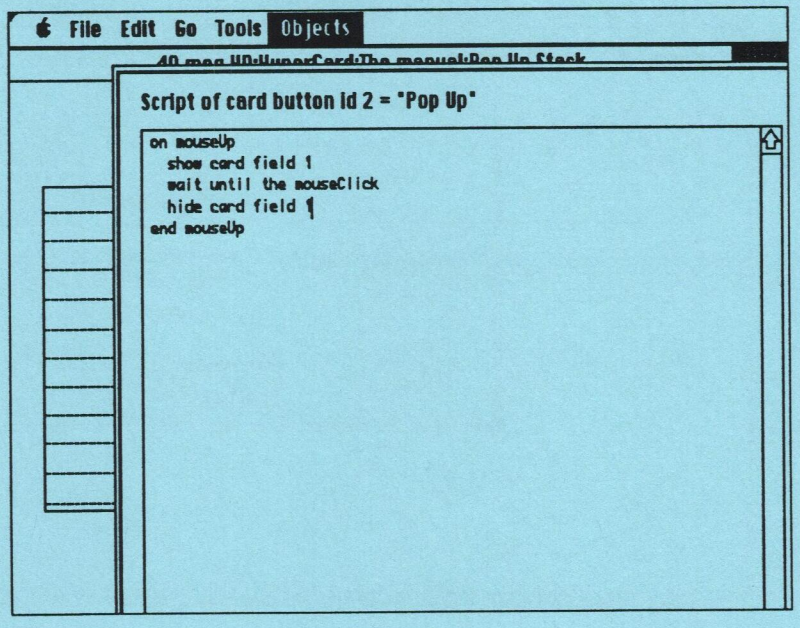


fig. 7 The Button script will cause the action on the screen. Note the wording of the wait statement. Type the script exactly as it appears here.

Hide	Show	Scripted Action	Event
A hypertalk command used to remove an object from view.	A hypertalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event which a script responds to.

Break

Drag Stack Chooses Paint Tools in the Script

After the break we will learn the final skill necessary for completion of the final project: the Ohio Stack. Before you can begin the Ohio Stack you need to know how to use the tool palette from script.

This will involve using the **choose** command and naming the tool from the palette. Each tool has a specific tool name. Within a script it is possible to choose a tool and change tools as often as needed. **(fig. 8)**

We will begin by making a new stack called **Drag Stack**. Complete a standard title card.

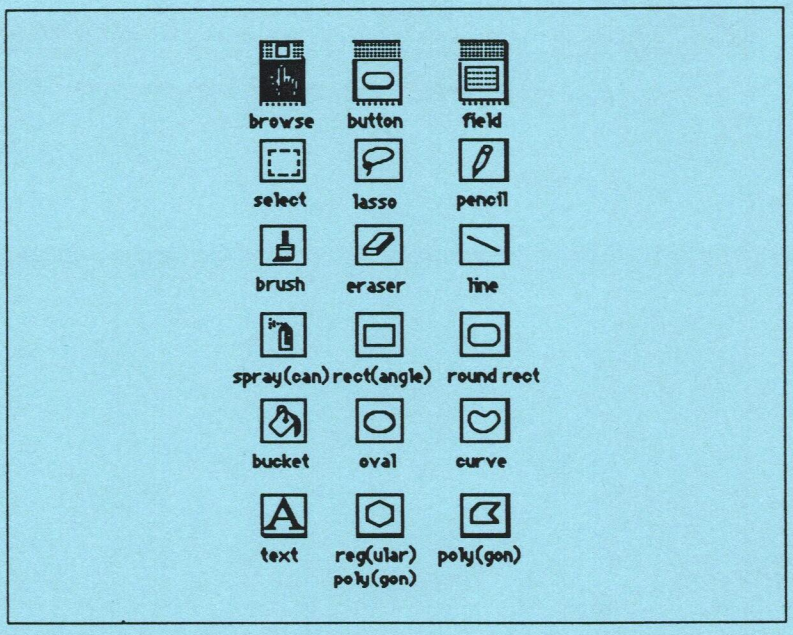


fig. 8 Each Tool on the palette is named specifically. The correct name must be used in a script to identify a tool.

Objects	Drag	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The command which causes a tool to be moved on a card.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Understanding the Size of the Screen

In order to drag a path from one spot on the screen to another, it will be necessary to understand the layout of the screen, which works like an ordinary rectangular coordinate system, almost...

Every point on the screen has an "address" composed of two numbers. The first number indicates how far the point is from the left margin and the second indicates the distance from the top of the screen. The units of measurement are pixels and the Macintosh screen is 512 pixels wide and 342 pixels down. **(fig. 9)**

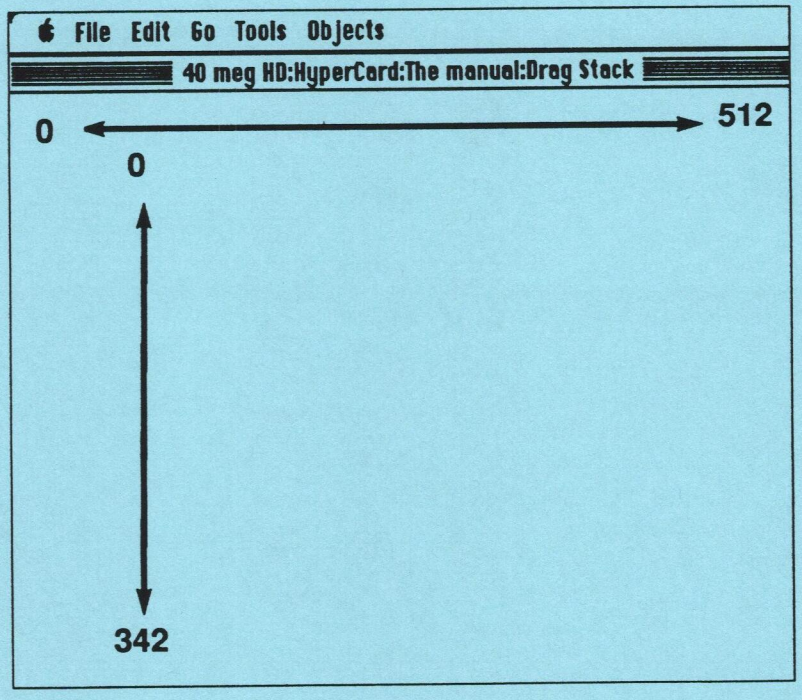


fig. 9 The Macintosh Screen is arranged in an invisible matrix of pixels: 512 across and 342 down. When scripting a point on the screen the horizontal number is named first and then the vertical.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Finding a "Spot" on the Screen

You can find the exact location of some point on the screen by opening the message box with **Command-M** and typing into the message box the words, "**the MouseLoc**" (short for mouse location). Hitting return will then return the exact present location of the mouse on the screen.

A pair of coordinates will come up in the message box, like 335,70, which would indicate that the cursor is presently located to the right of the center of the screen and fairly close to the top (about one fifth of the way down). This method of locating the cursor's coordinates takes some fussing, but gives you a handy way of recalling the dimensions of the screen. **(fig. 10)**

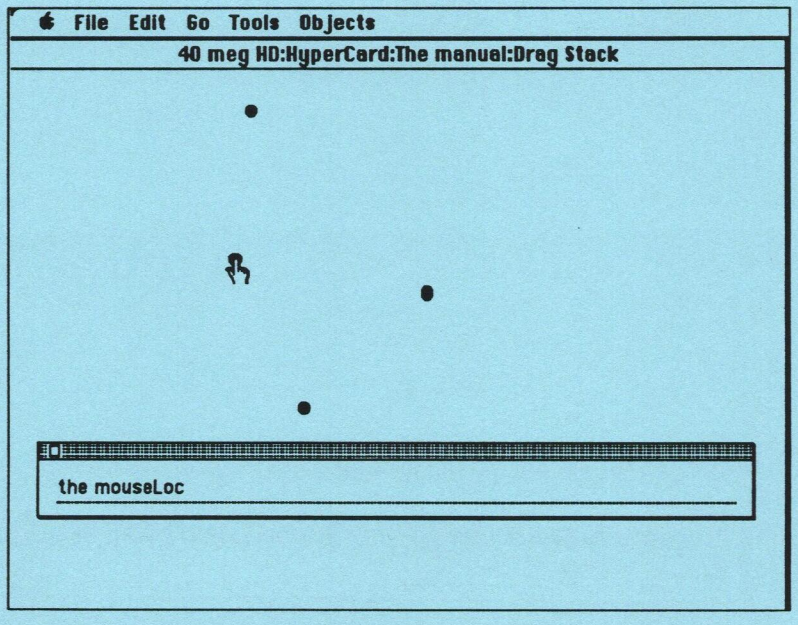


fig. 10 Exact locations on the screen may be found by typing "the mouseLoc" into the message box. Position the mouse (browse tool) at the desired spot and press the return key.

Objects	Choose	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The command to automatically select a tool from the menu.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Finding the Screen Dimensions

To recall the dimensions of the screen you can put the cursor in the upper left, get the MouseLoc and then repeat this operation with the cursor in the lower right. The coordinates in these two locations tell the basic dimensions of the screen, which can easily escape the minds of busy teachers, or anyone else. **(fig. 11)**

Experiment with the mouseLoc finding the coordinates of several "spots" on the screen.

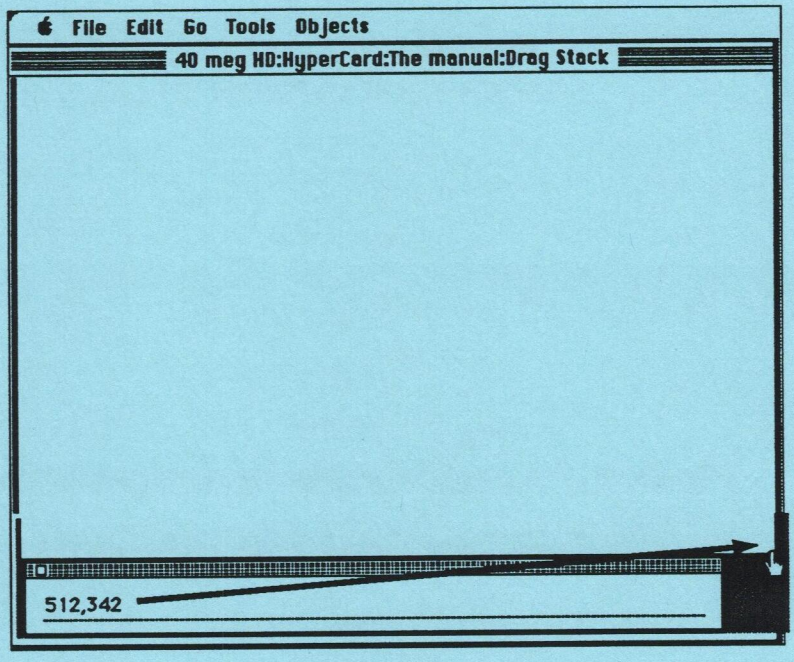


fig. 11 The complete dimensions of the screen may be determined by typing the mouseLoc into the message box and then move the mouse to the lower right corner of the screen. Press return and the horizontal and vertical coordinates appear in the message box.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Learning to Automate

With the measurements of the screen understood, the dragging of a line from one location to another is very simple:

choose _____ tool
drag from 15,50 to 475, 300

This scripting statement will cause something to be **dragged** from a point near the upper left of the screen to a point near the lower right. It is good to realize that the HyperTalk (HyperCard's scripting language) is very close to English. So if something is forgotten along the way, even attempting to write what should happen can be almost close enough to work.

Objects	Choose	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The command to automatically select a tool from the menu.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Changing Tools Within a Script

On the drag statement, something is dragged. But what? That depends on what you have selected before this line of code. A good way to handle this task is to drag the brush tool from one location to another. The script for the button "As the crow flies" accomplishes this:

```
on mouseUp
  choose brush tool
  drag from 15,50 to 475, 300
end mouseUp
```

(fig. 12)

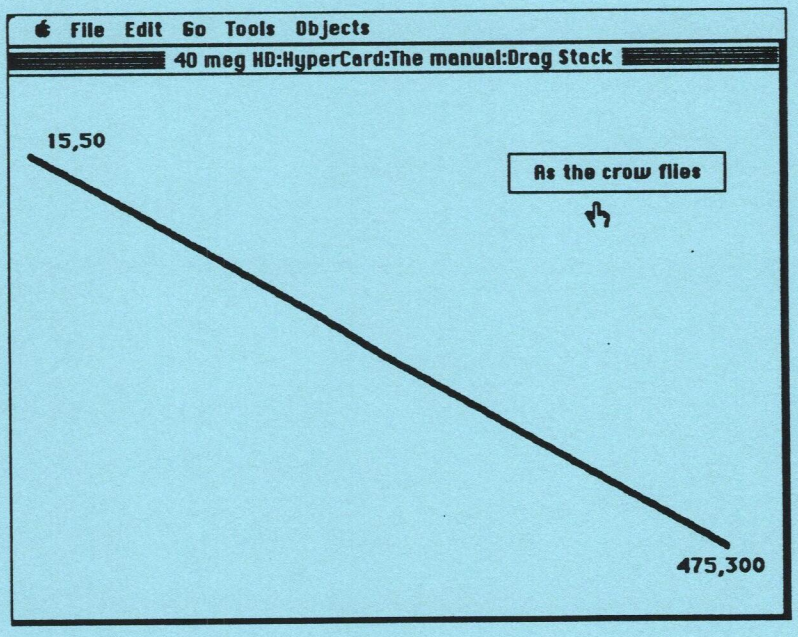


fig. 13 The drag command is used to drag the mouse from one location on the screen to another. This example demonstrates the script:

drag from 15,50 to 475,300.

The button "As the crow flies" activates this script.

Hide	Show	Scripted Action	Drag
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	The command which moves the cursor from one place to another.

The Completed Drag Statement

This script works fine except that it leaves the user still in the **brush tool mode** when the dragging job is done. To reset to the normal, browsing mode we simply add the line:

```
on Mouseup
  choose brush tool
  drag from 15,50 to 475, 300
  choose browse tool
end Mouseup
```

(fig. 13)

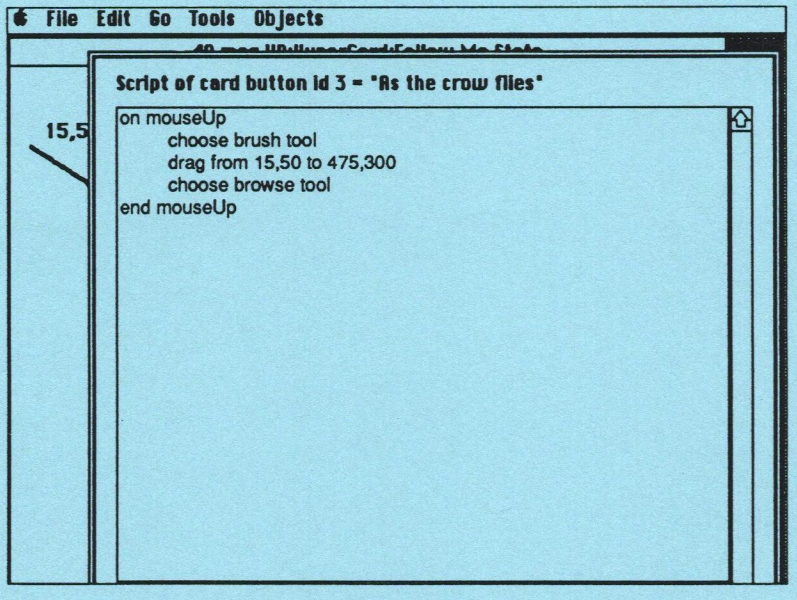


fig. 13 The completed command includes a return to the browse tool which readies the user to continue through the stack.

Objects	Hot Spot	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	The exact spot on the cursor where things happen.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Starting a State Stack

This completes the new ideas that you need to develop this little project. All of this demonstration, Follow-Me exercise should serve well as a reference for the work to come.

You may refer to the script or examples as you work. To do this you can simply use Recent to return to the stacks.

For now you will begin a **New Stack** naming it **Ohio Stack**. Take some time to complete the title card. Following the title card you will add a **New Background**.

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Drag
The command which moves the cursor from one place to another.

Outline the New Stack

The project calls for you to produce a stack that has three cards. The **first card** will be a title card similar to the title cards included in all of the other stacks.

The **second card** will have an outline map of the state of Ohio on a **new background**. On the card level there will be buttons representing five major cities in Ohio. When clicked these buttons will produce "pop up fields" with information on the cities they represent. The information will remain visible only until there is a click of the mouse anywhere on the screen. **(fig. 14)**

A **third card** will have the same outline map on the background with the same major cities but on this card an animated path from the "home" city to any other city will appear when a non-home city is "touched."

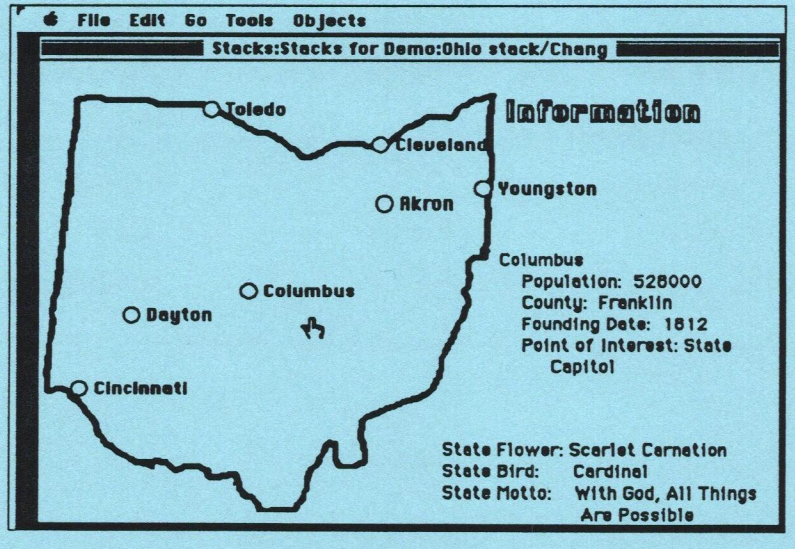


fig. 14 The Button for each city will cause a text field to appear containing information about that city.

Loc(ation)	Hot Spot	Handler	Connections
A command which returns the current location of the mouse.	The exact spot on the cursor where things happen.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Illustration of the Stack

For example, in Ohio the cities that information might be available for would be Cincinnati, Columbus, Cleveland, Youngstown, and Toledo. Any set of appropriate information could be used but the following information is suggested:

The county in which the city is found
The population of the city
The date of the founding of the city
A point of interest in that city

Clicking on a city button on card two would produce this information for the user.

Statistics of Ohio

City	County	Founding	Population
Cincinnati	Hamilton	1788	404,000
Cleveland	Cuyahoga	1796	660,000
Dayton	Montgomery	1796	219,000
Columbus	Franklin	1812	528,000
Toledo	Lucas	1833	366,000

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Set
The command which changes the property of an object or tool.

Card Three is Automated

On card three choosing Cincinnati as "home base" on the map would result in a path drawn from Cincinnati to any other city that the user would choose. **(fig. 15)**

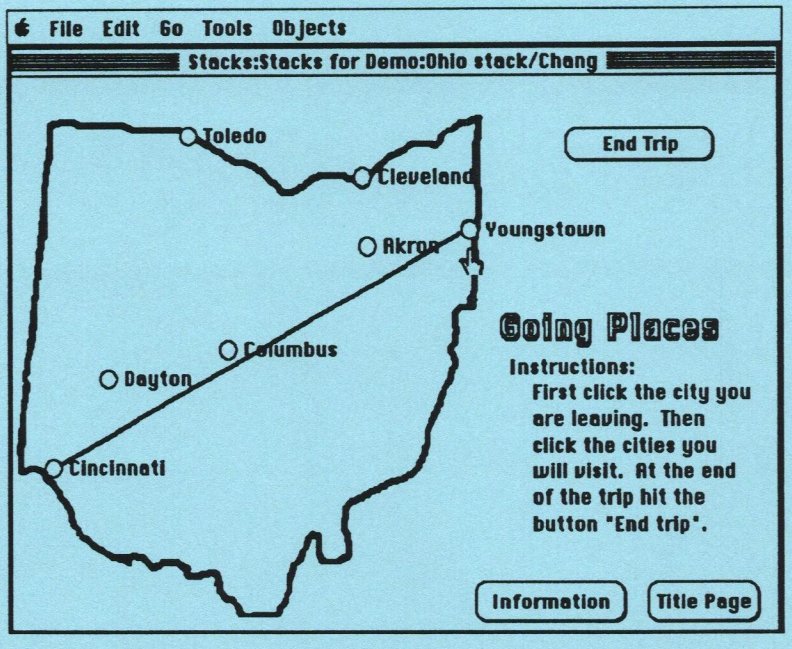


fig. 15 Card 3 of the Ohio Stack has buttons which will connect cities on the map showing a travel path between them. The path will be drawn automatically when the button is clicked.

Loc(ation)	Mode	Handler	Connections
A command which returns the current location of the mouse.	The mode defines action possible: browse, button, field, etc.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Preparing the Map with the Paint Tools

Once the title card is complete choose New Background on the Objects menu. Some review may be in order to recall the steps to Import a Paint file. You may need to look back at the process used to import Sunrise master in session 3.

The Ohio Map file is on the accompanying disk and should be imported into the background of card two. **(fig. 16)**

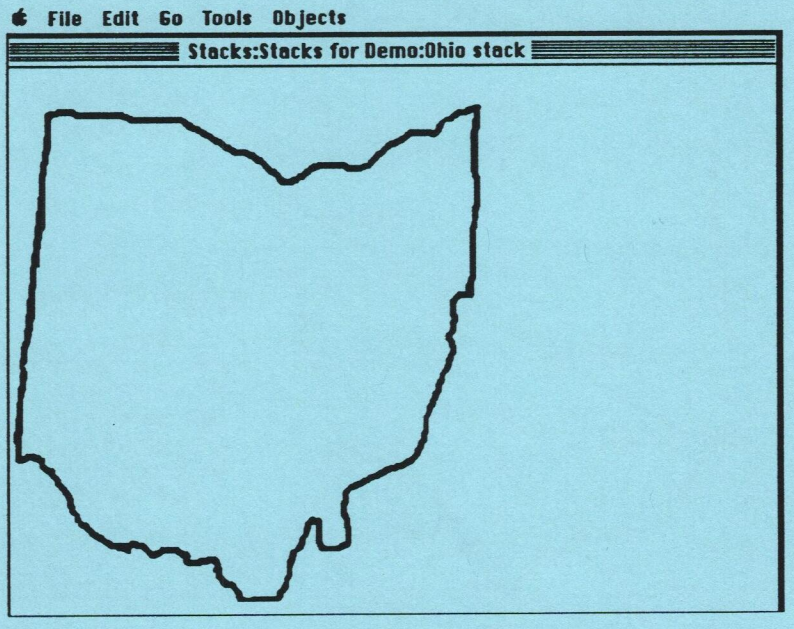


fig. 16 The outline map of Ohio will be imported into the background of the stack. A paint tool must be chosen in order to find the "Import Paint" option on the File menu. If the map is imported onto the card level by mistake it may be selected, cut, and pasted into the background.

Hide	Show	Scripted Action	Set
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	The command which changes the property of an object or tool.

Steps of this Project

To review, then, the steps in this project are:

1. Prepare a title card (this assumes they have already done a "scratch" card).

2. On the background of card two, import the outline of the state map (Ohio) from the disk that comes with this manual (or from your hard disk, where it is installed).

3. On the card level of card two, put in the names of the cities on long narrow buttons that include the black dot for the city. Script these buttons and fill in the information for each city on pop up fields.

Objects	the MouseLoc	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	A message to find the current coordinates of the mouse.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

4. On the card level of card three, put the same kind of buttons for the cities as on card two, but this time script the buttons so that when they are clicked a path from Cincinnati to that city is shown with a line on the card.

5. **If you get finished early, experiment with card three.**
   - **a.** Build another button and script it so that when it is clicked the paths that have been drawn are erased.
   - **b.** Experiment with the speed with which the path between the cities is drawn.
   - **c.** Vary the thickness of the path that is drawn between Cincinnati and the target city.
   - **d.** Change the script for the path buttons so that instead of always coming from Cincinnati, the path will be drawn from wherever the cursor is, so that a trip from Cincinnati to Cleveland is seen as the first leg of a trip that continues to Youngstown and then back to Cincinnati.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Conclusion of the Module

Review the important learnings:

The end of this module marks the end of the formal HyperCard training. HyperCard will be used and referred to in the Multi Media Module which may or may not accompany this training.

At the conclusion of this training the five objects of HyperCard should be identified: stack, background, card, button and field. All five have been created and modified during the four modules. You should be able to easily identify some characteristics of the objects.

Objects	the MouseLoc	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	A message to find the current coordinates of the mouse.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

The Principles of Navigation

The importance of navigation cannot be minimized. The lost feeling of not knowing where you are in a stack can be crippling to further use of HyperCard. You should be comfortable with the following ways to navigate:

1. **The Go menu**
2. **The keyboard equivalents of the Go menu**
3. **Scripted buttons**
4. **The Recent option on the Go menu**

(fig. 17)

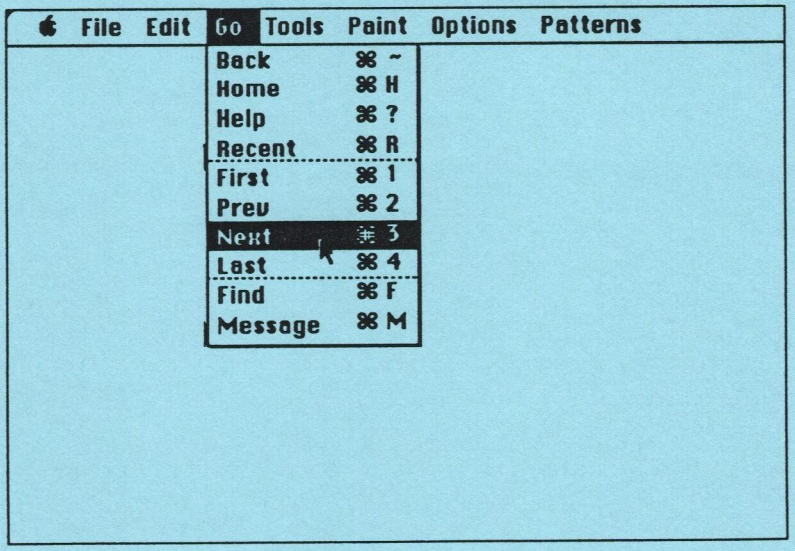


fig. 17 Navigation of a stack is critical to feeling confident with HyperCard. Four ways to navigate are the Go menu, keyboard commands, buttons and Recent.

Hide	Show	Scripted Action	Event
A HyperTalk command used to remove an object from view.	A HyperTalk command used to reveal graphics, text, or buttons.	Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.	Clicking the mouse is an event to which a script responds.

Other Skills Acquired

You have also learned to manage the **background** and **card level** of a stack. You have learned to write scripts for simple **navigation buttons** and more complex scripts to repeat a command more than once.

Drawings have been imported into a stack from other sources, both in and out of HyperCard. This involved the **Import Paint** command to capture an image from a MacPaint file. Images were moved between stacks by **copy**ing the image from one stack and **paste**ing it in another.

Objects	Navigation	Handler	Connections
Five objects are designed and scripted in Hyper-Card.	Movement through a stack possible in many ways.	Scripts are said to handle the events of the HyperCard objects.	HyperCard uses cards and buttons to make connections between ideas expressed in text, graphics and sound.

Creative Ideas and Empowering Students

Perhaps the most important aspect of this training is to help you grasp the power of HyperCard not for yourselves alone but for your students. It is not possible to learn "all there is to know" about HyperCard in two days.

What is possible is to glimpse the power of HyperCard to make a difference in education, empowering the student to use this revolutionary tool to make connections and, in some way, to create his own knowledge rather than to be the passive recipient of what others have created.

Hide
A HyperTalk command used to remove an object from view.

Show
A HyperTalk command used to reveal graphics, text, or buttons.

Scripted Action
Scripts may be written to simulate the action which could be done with the mouse by hand. Draw lines, write text or erase graphics at the click of the mouse.

Event
Clicking the mouse is an event to which a script responds.

HyperCard in Education

Survival Notes

Joseph F. Hofmeister • Joyce B. Rudowski
The HyperSchool
Cincinnati Country Day School
6905 Given Road, Cincinnati, Ohio 45243
513-561-7298 AppleLink K0197

Making a New Stack

1. Open HyperCard by either double clicking on the HyperCard application or an existing stack.
2. Choose the New Stack option on the File menu.
3. Type a name for the new stack in the box provided.
4. If you do not want to use the background of the current stack, click off the "Copy current background" choice.
5. Click on the New button and wait a few seconds for the first card of the new stack to appear.

Guide reference Module 1 page 15

Making a New Card

1. Choose the New Card option on the Edit menu or use the keyboard equivalent (command N).
2. A new card is added to the stack behind (following) the card currently visible.

Guide reference Module 1 page 22

Navigation

1. Navigating around a stack may be done using the Go menu. Choose the First, Prev(ious), Next, or Last options to move from one card to another.
2. The keyboard equivalents for the Go menu choices may be used: First- command-1, Prev- command-2, Next- command-3, and Last- command-4.
3. Buttons may be scripted to carry out the Go menu choices, as well as connecting cards not next to each other.
4. The Recent option on the Go menu provides quick navigation to cards previously visited.

Guide reference Module 1 page 26

Making a Field

1. Choose the field tool on the upper right side of the tool palette.
2. Choose the New Field option on the Objects menu.
3. With the field active (wiggling edges) select Field Info on the Objects menu.
4. The field may be moved on the card by holding down the arrow on the field and dragging the mouse.
5. The field may be resized by holding down the arrow on a corner of the field and dragging the mouse.

Guide reference Module 2 page 40

Entering Text into a Field

1. After making a new field choose the browse tool. The field may seem to disappear.
2. Roll the mouse across the card and watch for it to change from the browse hand to an I-bar.
3. Click the mouse when it changes to an I-bar.
4. Type the text you want on the field.

Guide reference Module 2 page 42

Duplicating Fields on a Card

1. Choose the field tool.
2. Hold down the Option key on the lower left side of the keyboard.
3. Click on the field you want to duplicate and holding down the mouse button, drag the mouse across the card.

Guide reference Module 2 page 41

Making a New Button

1. Choose the button tool, the middle tool on the top of the tool palette.
2. Choose the New Button option on the Objects menu.
3. With the button active (wiggling edges) select Button Info on the Objects menu.
4. The button may be moved on the card by selecting it. Then click on the button and drag the mouse.
5. The button may be resized by clicking on a corner of the button and dragging the mouse.

Guide reference Module 3 page 62

Changing the Style of a Button

1. Choose the button tool on the tool palette.
2. Click once on the button you want to change to select, or activate, the button.
3. Choose Button Info on the Objects menu and click on the choices you want to change; for example, transparent, radio, or round rectangle, show the name of the button or choose an Icon.

Guide reference Module 3 page 64

Writing the Script for a Button

1. Choose the button tool and double click on the button you want to script to get Button Info.
2. Click on "Script" to move to the script window.
3. Type in the script you want between the "on mouseUp" and "end mouseUp" which are provided.
4. Typical navigation scripts might be:

   go to next card
   go to prev card
   show all cards

Guide reference Module 3 page 67

Repeating in a Button Script

1. Design a button as desired.
2. Get Button Info by selecting the button and then choosing Button Info on the Objects menu.
3. Choose Script and type in the following:

```
repeat <a number> times
    go to next card
    wait <a number> ticks
end repeat
```

Guide reference Module 3 page 97

Scripting a Button to Hide and Show a Field (Pop Up Field)

1. Make a button and a field on a card. Check and make note of the field number on the Field Info for the field.
2. Go to the script of the button by choosing the button tool and getting Button Info for the button. Click on Script.
3. Type the following script between "on mouseUp" and "end mouseUp":

```
show card field <number>
wait until the mouseClick
hide card field <number>
```

Guide reference Module 4 page 119

Finding the Coordinates for a Spot on the Screen

1. Open the Message Box by choosing Message on the Go menu or using command-M on the keyboard.
2. Type "the mouseLoc" into the message box.
3. Move the mouse to the desired location on the screen and then press the Return key.
4. The numbers in the message box are the number of pixels across (first) and down (second) the screen to the chosen spot.

Guide reference Module 4 page 122

Scripting Ideas:

More Visual Effects

The following visual effects are available in HyperCard. The combinations of effects are numerous considering speed and color variations.

barn door (open or close) **wipe** (right, left, up or down)
scroll (right, left, up or down) **iris** (open or close)
dissolve **checkerboard**
zoom (in or out) **venetian blinds**

Speeds for the effects may include slow, very slow, fast and very fast.
The effect may include a *to* <*color*> command, using to black, to white, or to gray.
Examples of visual effects include:

visual effect wipe left slowly to white
visual effect dissolve fast to black
visual effect venetian blinds very slowly to gray

Assigning Random Numbers in HyperCard

Random numbers may be assigned to variables or card fields in HyperCard. The random number statement is

the random of <the upper limit>

The following statements assign (put) random numbers into card fields or variables:

put the random of 100 into card field 1
put the random of 2 into heads
put the random of 6 into dice

The lower limit of the random statement is assumed to be 1.

The lower limit may be changed by adding to the random statement as follows:

put the random of 100 + 50 into card field 1

This statement causes the random selection of a number from 1 to 100 and then adds 50 to the number.

Shortcuts

1. Move to the background of a card using **command-B.**
2. Show or hide the menu bar using **command-space bar.**
3. When the browse tool is chosen you may go directly to the script of a button by holding down the **option** and **command** keys and **click once** on the button.
4. Transparent buttons may be seen on a card by holding down the **option** and **command** keys at the same time.
5. **Double clicking** on a button while using the button tool will get the information for that button. Similarly, double clicking on a field with the field tool will get Field Info.
6. Both the **Tools and Patterns menus** may be "torn off" the menu bar by dragging the mouse all the way down the menu to a place on the screen. The menus may be "put away" by clicking in the small box in the upper left corner of the menu.
7. **Double clicking** on the paint tools will provide a shortcut to a property of the tool. For example, double clicking on the eraser erases all the graphics on the card, or double clicking on the oval changes it from unfilled to filled. Double clicking on the regular polygon reveals the shape choices for the regular polygon tool.